# 32-bit Arm® Cortex®-M4F MCUs with 1 Msps 12-bit ADC, QSPI, USB, Ethernet, and PTC

## SAM D5x/E5x Family

Product Page Links

## Features

### Operating Conditions

- 1.71V to 3.63V, -40°C to +125°C, DC to 100 MHz
- 1.71V to 3.63V, -40°C to +105°C, DC to 120 MHz
- 1.71V to 3.63V, -40°C to +85°C, DC to 120 MHz

### Core: 120 MHz Arm Cortex-M4

- 403 CoreMark® at 120 MHz
- 4 KB combined instruction cache and data cache
- 8-zone Memory Protection Unit (MPU)
- Thumb®-2 instruction set
- Embedded Trace Module (ETM) with instruction trace stream
- CoreSight™ Embedded Trace Buffer (ETB)
- Trace Port Interface Unit (TPIU)
- Floating Point Unit (FPU)

### Memories

- 1 MB, 512 KB, or 256 KB in-system self-programmable (ISSP) Flash with:
  - Error Correction Code (ECC)
  - Dual bank with read-while-write (RWW) support
  - EEPROM hardware emulation (SmartEEPROM)
- 128 KB, 192 KB, 256 KB SRAM main memory
  - 64 KB, 96 KB, 128 KB of Error Correction Code (ECC) RAM option
- Up to 4 KB of Tightly Coupled Memory (TCM)
- Up to 8 KB additional SRAM
  - Can be retained in Backup mode
- Eight 32-bit Backup registers

### System

- Power-on Reset (POR) and Brown-out detection (BOD)
- Internal and external clock options
- External Interrupt Controller (EIC)
- 16 external interrupts
- One non-maskable interrupt
- Two-pin Serial Wire Debug (SWD) programming, test, and debugging interface

### Power Supply


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

- Idle, Standby, Hibernate, Backup, and Off sleep modes
- SleepWalking peripherals
- Battery backup support
- Embedded Buck/LDO regulator supporting on-the-fly selection

**High-Performance Peripherals**

- 32-channel Direct Memory Access Controller (DMAC)
  - Built-in CRC with memory CRC generation/monitor hardware support
- Up to two SD/MMC Host Controller (SDHC)
  - Up to 50 MHz operation
  - 4-bit or 1-bit interface
  - Compatibility with SD and SDHC memory card specification version 3.01
  - Compatibility with SDIO specification version 3.0
  - Compliant with JDEC specification, MMC memory cards V4.51
- One Quad I/O Serial Peripheral Interface (QSPI)
  - eXecute-In-Place (XIP) support
  - Dedicated AHB memory zone
- One Ethernet MAC (SAM E53 and SAM E54)
  - 10/100 Mbps in MII and RMII with dedicated DMA
  - IEEE® 1588 Precision Time Protocol (PTP) support
  - IEEE 1588 Time Stamping Unit (TSU) support
  - IEEE 802.3AZ energy efficiency support
  - Support for 802.1AS and 1588 precision clock synchronization protocol
  - Wake on LAN support
- Up to two Controller Area Network (CAN) (that is., SAM E51 and SAM E54)
  - Support for CAN 2.0A/CAN 2.0B and CAN-FD (ISO 11898-1:2015)
- One Full-Speed (12 Mbps) Universal Serial Bus (USB) 2.0 interface:
  - Embedded host and device function
  - Eight endpoints
  - On-chip transceiver with integrated serial resistor

**System Peripherals**

- 32-channel Event System
- Up to eight Serial Communication Interfaces (SERCOM), each configurable to operate as either:
  - USART with full-duplex and single-wire half-duplex configuration
  - ISO7816
  - $I^2C$ up to 3.4 MHz
  - SPI
  - LIN Host/Client
  - RS485
  - SPI inter-byte space
- Up to eight 16-bit Timers/Counters (TC) each configurable as:
  - 16-bit TC with two compare/capture channels


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - 8-bit TC with two compare/capture channels
  - 32-bit TC with two compare/capture channels, by pairing two TCs
- Two 24-bit Timer/Counters for Control (TCC) with extended functions:
  - Up to six compare channels with optional complementary output
  - Generation of synchronized Pulse-Width Modulation (PWM) pattern across port pins
  - Deterministic fault protection, fast decay and configurable dead-time between complementary output
  - Dithering that increase resolution with up to 5 bit and reduce quantization error
- Up to Three 16-bit Timer/Counters for Control (TCC) with extended functions:
  - Up to three compare channels with optional complementary output
- PWM Modes using TC and TCC peripherals:
  - Up to six PWM channels on each 24-bit TCC
  - Up to three PWM channels on each 16-bit TCC
  - Up to two PWM channels on each 16-bit TC
- 32-bit Real Time Counter (RTC) with clock/calendar function
- Up to 5 wake-up pins with tamper detection and debouncing filter
- Watchdog Timer (WDT) with Window mode
- CRC-32 generator
- One two-channel Inter-IC Sound Interface ($I^2S$)
- Position Decoder (PDEC)
- Frequency meter (FREQM)
- One Four-LUTs Configurable Custom Logic (CCL)
- Dual 12-bit, 1 MSPS Analog-to-Digital Converter (ADC) with up to 16 channels each:
  - Differential-ended and single-ended input
  - Automatic offset and gain error compensation
  - Oversampling and decimation in hardware to support 13-bit, 14-bit, 15-bit, or 16-bit resolution
- Dual 12-bit, 1 MSPS output Digital-to-Analog Converter (DAC)
- Two Analog Comparators (AC) with Window Compare function
- Two temperature sensors
- Parallel Capture Controller (PCC)
  - Up to 14-bit parallel capture mode
- Peripheral Touch Controller (PTC)
  - Capacitive Touch buttons, sliders, and wheels
  - Wake-up on touch
  - Up to 32 self-capacitance and up to 256 mutual-capacitance channels

**Cryptography**

- One Advanced Encryption System (AES) with 256-bit key length and up to 2 MB/s data rate
  - Five confidential modes of operation: ECB, CBC, CFB, OFB, CTR
  - Supports counter with CBC-MAC mode
  - Galois Counter Mode (GCM)
- True Random Number Generator (TRNG)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Public Key Cryptography Controller (PUKCC) and associated Classical Public Key Cryptography Library (PUKCL)
  - RSA, DSA
  - Elliptic Curves Cryptography (ECC) ECC GF(2n), ECC GF(p)
- Integrity Check Module (ICM) based on Secure Hash Algorithm (SHA1, SHA224, SHA256), DMA assisted

### Oscillators

- 32.768 kHz crystal oscillator (XOSC32K)
  - Clock failure detection
- Up to two 8 MHz to 48 MHz crystal oscillator (XOSC)
  - Clock failure detection
- 32.768 kHz ultra-low power internal oscillator (OSCULP32K)
- 48 MHz Digital Frequency Locked Loop (DFLL48M)
- Two 96-200 MHz Fractional Digital Phased Locked Loop (FDPLL200M)

### I/O

- Up to 99 programmable I/O pins

### Qualification

- AEC-Q100 Grade 1 (-40°C to 125°C)

### Related Devices

Products that support the same set of applications and are compatible with the SAM D5x/E5x Family and provide expanded options:

- The PIC32CX SG41/SG60/SG61 family of devices (*DS60001715*)
  - Pinout and Software compatible to the SAMD5x/E5x family
  - Enhanced security capabilities:
    - Immutable secure boot and secure debug
    - Options with integrated Hardware Security Module (HSM)
- The PIC32CK SG01/SG00/GC01/GC00 family of devices (*DS60001795*)
  - Roadmap Solution compatible with SAMD5x/E5x
  - Offers expanded Flash/RAM memory up to 2MB Flash/512KB RAM
  - Enhanced security capabilities:
    - Immutable secure boot and secure debug
    - Options with integrated Hardware Security Module (HSM)
  - Up to two USB ports one high-speed and one full-speed
  - Improved Peripheral Touch Controller (PTC)

### Packages

**Table 1.** Package Types

| Parameter | Package Type | | | | | | |
|---|---|---|---|---|---|---|---|
| | VQFN | | TQFP | | | TFBGA | WLCSP |
| Pin Count | 48 | 64 | 64 | 100 | 128 | 120 | 64 |
| I/O Pins (up to) | 37 | 51 | 51 | 81 | 99 | 99 | 51 |
| Contact/Lead Pitch | 0.5 | 0.5 | 0.5 | 0.5 | 0.4 | 0.5 | 0.4 |
| Dimension | 7x7x0.9 | 9x9x0.9 | 10x10x1.2 | 14x14x1.2 | 14x14x1.2 | 8x8x1.2 | 3.59x3.51x0.53 |

**Note:** All dimensions are in millimeter (mm) unless specified.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# Table of Contents

Features ... 1

1. Configuration Summary ... 17
2. Ordering Information ... 19
3. Block Diagram ... 20
   - 3.1. SAM D5x/E5x Block Diagram ... 20
4. Pinout ... 21
   - 4.1. 48-Pin VQFN Package ... 21
   - 4.2. 64-Pin TQFP and VQFN Package ... 22
   - 4.3. 64-Pin WLCSP Package ... 23
   - 4.4. 100-Pin TQFP Package ... 24
   - 4.5. 120-Ball TFBGA Package ... 25
   - 4.6. 128-Pin TQFP Package ... 26
5. Signal Descriptions List ... 27
6. I/O Multiplexing and Considerations ... 30
   - 6.1. Multiplexed Signals ... 30
   - 6.2. Other Functions ... 34
7. Power Supply and Start-Up Considerations ... 43
   - 7.1. Power Domain Overview ... 43
   - 7.2. Power Supply Considerations ... 43
   - 7.3. Power-Up ... 45
   - 7.4. Power-On Reset and Brown-Out Detector ... 46
8. Product Memory Mapping Overview ... 48
9. Memories ... 50
   - 9.1. Embedded Memories ... 50
   - 9.2. Physical Memory Map ... 50
   - 9.3. SRAM Memory Configuration ... 50
   - 9.4. NVM User Row Mapping (UROW) ... 53
   - 9.5. NVM Software Calibration Area Mapping (SW0) ... 54
   - 9.6. Serial Number ... 55
10. Processor and Architecture ... 56
    - 10.1. Cortex M4 Processor ... 56
    - 10.2. Nested Vector Interrupt Controller ... 59
    - 10.3. High-Speed Bus System ... 68
11. CMCC - Cortex M Cache Controller ... 71
    - 11.1. Overview ... 71
    - 11.2. Features ... 71
    - 11.3. Block Diagram ... 72
    - 11.4. Signal Description ... 73
    - 11.5. Product Dependencies ... 73


© 2026 Microchip Technology Inc. and its subsidiaries
MICROCHIP
Downloaded from Arrow.com.

11.6. Functional Description ... 73
11.7. DEBUG Mode ... 76
11.8. RAM Properties ... 76
11.9. Register Summary ... 78
11.10. Register Description ... 78

12. DSU - Device Service Unit ... 91
12.1. Overview ... 91
12.2. Features ... 91
12.3. Block Diagram ... 91
12.4. Signal Description ... 92
12.5. Product Dependencies ... 92
12.6. Debug Operation ... 93
12.7. Chip Erase ... 95
12.8. Programming ... 95
12.9. Intellectual Property Protection ... 96
12.10. Device Identification ... 97
12.11. Functional Description ... 98
12.12. Register Summary ... 103
12.13. Register Description ... 105

13. Clock System ... 133
13.1. Clock Distribution ... 133
13.2. Synchronous and Asynchronous Clocks ... 134
13.3. Register Synchronization ... 135
13.4. Enabling a Peripheral ... 137
13.5. On Demand Clock Requests ... 138
13.6. Power Consumption vs. Speed ... 138
13.7. Clocks after Reset ... 138

14. GCLK - Generic Clock Controller ... 140
14.1. Overview ... 140
14.2. Features ... 140
14.3. Block Diagram ... 140
14.4. Signal Description ... 141
14.5. Product Dependencies ... 141
14.6. Functional Description ... 142
14.7. Register Summary ... 148
14.8. Register Description ... 148

15. MCLK – Main Clock ... 157
15.1. Overview ... 157
15.2. Features ... 157
15.3. Block Diagram ... 157
15.4. Signal Description ... 157
15.5. Product Dependencies ... 157
15.6. Functional Description ... 159
15.7. Register Summary ... 164
15.8. Register Description ... 164


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

16. RSTC – Reset Controller ..... 181
16.1. Overview ..... 181
16.2. Features ..... 181
16.3. Block Diagram ..... 181
16.4. Signal Description ..... 181
16.5. Product Dependencies ..... 181
16.6. Functional Description ..... 182
16.7. Register Summary ..... 185
16.8. Register Description ..... 185

17. RAMECC – RAM Error Correction Code (ECC) ..... 188
17.1. Overview ..... 188
17.2. Features ..... 188
17.3. Block Diagram ..... 188
17.4. Signal Description ..... 188
17.5. Product Dependencies ..... 188
17.6. Functional Description ..... 190
17.7. Register Summary ..... 191
17.8. Register Description ..... 191

18. PM – Power Manager ..... 198
18.1. Overview ..... 198
18.2. Features ..... 198
18.3. Block Diagram ..... 198
18.4. Signal Description ..... 198
18.5. Product Dependencies ..... 198
18.6. Functional Description ..... 199
18.7. Register Summary ..... 208
18.8. Register Description ..... 208

19. SUPC – Supply Controller ..... 218
19.1. Overview ..... 218
19.2. Features ..... 218
19.3. Block Diagram ..... 219
19.4. Signal Description ..... 219
19.5. Product Dependencies ..... 219
19.6. Functional Description ..... 221
19.7. Register Summary ..... 228
19.8. Register Description ..... 228

20. WDT – Watchdog Timer ..... 247
20.1. Overview ..... 247
20.2. Features ..... 247
20.3. Block Diagram ..... 247
20.4. Signal Description ..... 248
20.5. Product Dependencies ..... 248
20.6. Functional Description ..... 249
20.7. Register Summary ..... 254
20.8. Register Description ..... 254


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

21. RTC – Real-Time Counter....263
21.1. Overview....263
21.2. Features....263
21.3. Block Diagram....263
21.4. Signal Description....264
21.5. Product Dependencies....264
21.6. Functional Description....266
21.7. Register Summary - Mode 0 - 32-Bit Counter....277
21.8. Register Description - Mode 0 - 32-Bit Counter....278
21.9. Register Summary - Mode 1 - 16-Bit Counter....300
21.10. Register Description - Mode 1 - 16-Bit Counter....301
21.11. Register Summary - Mode 2 - Clock/Calendar....324
21.12. Register Description - Mode 2 - Clock/Calendar....325

22. DMAC – Direct Memory Access Controller....348
22.1. Overview....348
22.2. Features....348
22.3. Block Diagram....350
22.4. Signal Description....350
22.5. Product Dependencies....350
22.6. Functional Description....351
22.7. Register Summary....376
22.8. Register Description....384
22.9. Register Summary - SRAM....413
22.10. Register Description - SRAM....413

23. EIC – External Interrupt Controller....420
23.1. Overview....420
23.2. Features....420
23.3. Block Diagram....420
23.4. Signal Description....420
23.5. Product Dependencies....421
23.6. Functional Description....422
23.7. Register Summary....429
23.8. Register Description....429

24. GMAC - Ethernet MAC....445
24.1. Description....445
24.2. Features....445
24.3. Block Diagram....446
24.4. Signal Description....446
24.5. Product Dependencies....446
24.6. Functional Description....447
24.7. Programming Interface....471
24.8. Register Summary....475
24.9. Register Description....482

25. NVMCTRL – Nonvolatile Memory Controller....598
25.1. Overview....598
25.2. Features....598


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

25.3. Block Diagram ..... 598
25.4. Signal Description ..... 599
25.5. Product Dependencies ..... 599
25.6. Functional Description ..... 600
25.7. Register Summary ..... 619
25.8. Register Description ..... 619

26. ICM - Integrity Check Monitor ..... 641
26.1. Overview ..... 641
26.2. Features ..... 641
26.3. Block Diagram ..... 642
26.4. Signal Description ..... 642
26.5. Product Dependencies ..... 642
26.6. Functional Description ..... 643
26.7. Register Summary - ICM ..... 657
26.8. Register Description ..... 658

27. PAC - Peripheral Access Controller ..... 676
27.1. Overview ..... 676
27.2. Features ..... 676
27.3. Block Diagram ..... 676
27.4. Product Dependencies ..... 676
27.5. Functional Description ..... 677
27.6. Register Summary ..... 681
27.7. Register Description ..... 681

28. OSCCTRL – Oscillators Controller ..... 712
28.1. Overview ..... 712
28.2. Features ..... 712
28.3. Block Diagram ..... 713
28.4. Signal Description ..... 713
28.5. Product Dependencies ..... 713
28.6. Functional Description ..... 714
28.7. Register Summary ..... 727
28.8. Register Description ..... 728

29. OSC32KCTRL – 32KHz Oscillators Controller ..... 755
29.1. Overview ..... 755
29.2. Features ..... 755
29.3. Block Diagram ..... 755
29.4. Signal Description ..... 756
29.5. Product Dependencies ..... 756
29.6. Functional Description ..... 757
29.7. Register Summary ..... 762
29.8. Register Description ..... 762

30. FREQM – Frequency Meter ..... 773
30.1. Overview ..... 773
30.2. Features ..... 773
30.3. Block Diagram ..... 773


© 2026 Microchip Technology Inc. and its subsidiaries
MICROCHIP
Downloaded from Arrow.com.

30.4. Signal Description....773
30.5. Product Dependencies....773
30.6. Functional Description....775
30.7. Register Summary....778
30.8. Register Description....778

31. Event System (EVSYS)....788
31.1. Overview....788
31.2. Features....788
31.3. Block Diagram....788
31.4. Product Dependencies....789
31.5. Functional Description....790
31.6. Register Summary....797
31.7. Register Description....801

32. PORT - I/O Pin Controller....819
32.1. Overview....819
32.2. Features....819
32.3. Block Diagram....820
32.4. Signal Description....820
32.5. Product Dependencies....820
32.6. Functional Description....822
32.7. Register Summary....828
32.8. Register Description....829

33. SERCOM – Serial Communication Interface....848
33.1. Overview....848
33.2. Features....848
33.3. Block Diagram....849
33.4. Signal Description....849
33.5. Product Dependencies....849
33.6. Functional Description....851

34. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter....856
34.1. Overview....856
34.2. USART Features....856
34.3. Block Diagram....857
34.4. Signal Description....857
34.5. Product Dependencies....857
34.6. Functional Description....859
34.7. Register Summary....876
34.8. Register Description....876

35. SERCOM SPI – SERCOM Serial Peripheral Interface....903
35.1. Overview....903
35.2. Features....903
35.3. Block Diagram....904
35.4. Signal Description....904
35.5. Product Dependencies....904
35.6. Functional Description....906


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

35.7. Register Summary ..... 917
35.8. Register Description ..... 917

36. SERCOM I²C – Inter-Integrated Circuit ..... 938
36.1. Overview ..... 938
36.2. Features ..... 938
36.3. Block Diagram ..... 939
36.4. Signal Description ..... 939
36.5. Product Dependencies ..... 939
36.6. Functional Description ..... 941
36.7. Register Summary - I2C Client ..... 960
36.8. Register Description - I²C Client ..... 960
36.9. Register Summary - I2C Host ..... 977
36.10. Register Description - I²C Host ..... 977

37. QSPI - Quad Serial Peripheral Interface ..... 996
37.1. Overview ..... 996
37.2. Features ..... 996
37.3. Block Diagram ..... 997
37.4. Signal Description ..... 997
37.5. Product Dependencies ..... 997
37.6. Functional Description ..... 999
37.7. Register Summary ..... 1016
37.8. Register Description ..... 1017

38. USB – Universal Serial Bus ..... 1038
38.1. Overview ..... 1038
38.2. Features ..... 1038
38.3. USB Block Diagram ..... 1039
38.4. Signal Description ..... 1039
38.5. Product Dependencies ..... 1039
38.6. Functional Description ..... 1041
38.7. Common Device / Host Register Summary ..... 1060
38.8. General Device Register Summary ..... 1067
38.9. Device Endpoint n Register Summary ..... 1080
38.10. Endpoint Descriptors Structure ..... 1090
38.11. Device Endpoint n Descriptor Bank 0/1 ..... 1091
38.12. General Host Register Summary ..... 1097
38.13. Host Pipe n Register Summary ..... 1111
38.14. Pipe Descriptors Structure ..... 1123
38.15. Host Pipe n Descriptor Bank 0 / 1 ..... 1124

39. CAN - Control Area Network ..... 1133
39.1. Overview ..... 1133
39.2. Features ..... 1133
39.3. Block Diagram ..... 1134
39.4. Signal Description ..... 1134
39.5. Product Dependencies ..... 1134
39.6. Functional Description ..... 1135
39.7. Register Summary ..... 1155


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

39.8. Register Description....1158
39.9. Message RAM....1225

40. SD/MMC Host Controller (SDHC)....1234
40.1. Overview....1234
40.2. Features....1234
40.3. Block Diagrams....1235
40.4. Signal Description....1236
40.5. Product Dependencies....1236
40.6. Functional Description....1236
40.7. Register Summary....1238
40.8. Register Description....1240

41. Configurable Custom Logic (CCL)....1310
41.1. Overview....1310
41.2. Features....1310
41.3. Block Diagram....1311
41.4. Signal Description....1311
41.5. Product Dependencies....1311
41.6. Functional Description....1312
41.7. Register Summary....1323
41.8. Register Description....1323

42. AES – Advanced Encryption Standard....1328
42.1. Overview....1328
42.2. Features....1328
42.3. Block Diagram....1329
42.4. Signal Description....1330
42.5. Product Dependencies....1330
42.6. Functional Description....1331
42.7. Register Summary....1340
42.8. Register Description....1342

43. Public Key Cryptography Controller (PUKCC)....1359
43.1. Overview....1359
43.2. Product Dependencies....1359
43.3. Functional Description....1360

44. TRNG – True Random Number Generator....1470
44.1. Overview....1470
44.2. Features....1470
44.3. Block Diagram....1470
44.4. Signal Description....1470
44.5. Product Dependencies....1470
44.6. Functional Description....1471
44.7. Register Summary....1474
44.8. Register Description....1474

45. ADC – Analog-to-Digital Converter....1481
45.1. Overview....1481


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

45.2. Features....1481
45.3. Block Diagram....1482
45.4. Signal Description....1482
45.5. Product Dependencies....1483
45.6. Functional Description....1484
45.7. Register Summary....1500
45.8. Register Description....1501

46. AC – Analog Comparators....1532
46.1. Overview....1532
46.2. Features....1532
46.3. Block Diagram....1533
46.4. Signal Description....1533
46.5. Product Dependencies....1533
46.6. Functional Description....1535
46.7. Register Summary....1545
46.8. Register Description....1545

47. DAC – Digital-to-Analog Converter....1562
47.1. Overview....1562
47.2. Features....1562
47.3. Block Diagram....1562
47.4. Signal Description....1562
47.5. Product Dependencies....1563
47.6. Functional Description....1564
47.7. Register Summary....1575
47.8. Register Description....1575

48. Timer/Counter (TC)....1600
48.1. Overview....1600
48.2. Features....1600
48.3. Block Diagram....1601
48.4. Signal Description....1601
48.5. Product Dependencies....1602
48.6. Functional Description....1603
48.7. Register Description....1619

49. TCC – Timer/Counter for Control Applications....1682
49.1. Overview....1682
49.2. Features....1682
49.3. Block Diagram....1683
49.4. Signal Description....1683
49.5. Product Dependencies....1684
49.6. Functional Description....1685
49.7. Register Summary....1719
49.8. Register Description....1721

50. PTC - Peripheral Touch Controller....1760
50.1. Overview....1760
50.2. Features....1760


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

50.3. Block Diagram....1761
50.4. Signal Description....1761
50.5. System Dependencies....1762
50.6. Functional Description....1763

51. Inter-IC ($I^2S$) Sound Controller....1764
51.1. Overview....1764
51.2. Features....1764
51.3. Block Diagram....1765
51.4. Signal Description....1765
51.5. Product Dependencies....1766
51.6. Functional Description....1767
51.7. $I^2S$ Application Examples....1778
51.8. Register Summary....1781
51.9. Register Description....1781

52. PCC - Parallel Capture Controller....1799
52.1. Overview....1799
52.2. Features....1799
52.3. Block Diagram....1799
52.4. Signal Description....1799
52.5. Product Dependencies....1799
52.6. Functional Description....1801
52.7. Register Summary....1807
52.8. Register Description....1807

53. PDEC – Position Decoder....1817
53.1. Overview....1817
53.2. Features....1817
53.3. Block Diagram....1818
53.4. Signal Description....1818
53.5. Product Dependencies....1818
53.6. Functional Description....1820
53.7. Register Summary....1832
53.8. Register Description....1832

54. Electrical Characteristics at 85°C....1858
54.1. Disclaimer....1858
54.2. Absolute Maximum Ratings....1858
54.3. General Operating Ratings....1858
54.4. Injection Current....1859
54.5. Supply Characteristics....1860
54.6. Maximum Clock Frequencies....1860
54.7. Power Consumption....1861
54.8. Wake-Up Time....1864
54.9. I/O Pin Characteristics....1865
54.10. Analog Characteristics....1866
54.11. PTC Characteristics....1878
54.12. NVM Characteristics....1879
54.13. Oscillators Characteristics....1880


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

54.14. Timing Characteristics ... 1885
54.15. USB Characteristics ... 1902

55. Electrical Characteristics at 105°C ... 1903
55.1. General Operating Ratings (105°C) ... 1903
55.2. Power Consumption (105°C) ... 1903
55.3. Analog Characteristics (105°C) ... 1906
55.4. NVM Characteristics ... 1916
55.5. Oscillators Characteristics (105°C) ... 1916

56. Electrical Characteristics at 125°C ... 1919
56.1. General Operating Ratings (125°C) ... 1919
56.2. Injection Current (125°C) ... 1919
56.3. Maximum Clock Frequencies (125°C) ... 1919
56.4. Power Consumption (125°C) ... 1920
56.5. Analog Characteristics (125°C) ... 1923
56.6. NVM Characteristics (125°C) ... 1933
56.7. Oscillators Characteristics (125°C) ... 1934
56.8. Timing Characteristics (125°C) ... 1935

57. AEC Q-100 Grade 1, 125°C Electrical Characteristics ... 1936

58. Appendix A ... 1937
58.1. ISELED FULL License Enabled Functional Devices ... 1937
58.2. Ordering Information ... 1937

59. Packaging Information ... 1938
59.1. Package Marking Information ... 1938
59.2. Thermal Considerations ... 1938
59.3. Package Drawings ... 1939
59.4. Soldering Profile ... 1965

60. Schematic Checklist ... 1966
60.1. Introduction ... 1966
60.2. Power Supply ... 1966
60.3. External Analog Reference Connections ... 1968
60.4. External Reset Circuit ... 1971
60.5. Unused or Unconnected Pins ... 1972
60.6. Clocks and Crystal Oscillators ... 1972
60.7. Programming and Debug Ports ... 1975
60.8. QSPI Interface ... 1978
60.9. USB Interface ... 1978
60.10. SDHC Interface ... 1979

61. Conventions ... 1980
61.1. Numerical Notation ... 1980
61.2. Memory Size and Type ... 1980
61.3. Frequency and Time ... 1980
61.4. Registers and Bits ... 1980

62. Acronyms and Abbreviations ... 1982


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

63. Product Identification System ..... 1984

64. Revision History ..... 1985

Microchip Information ..... 2001

Trademarks ..... 2001

Legal Notice ..... 2001

Microchip Devices Code Protection Feature ..... 2001

Product Page Links ..... 2002


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

# 1. Configuration Summary

**Table 1-1.** SAM E53/E54 Family Features with Ethernet

| Device | Flash (KB) | SRAM (KB) / Backup RAM (KB) | Pins | Packages | Peripherals | | | | | | | | | | | | | | | | | | | | Analog | | | | | Security | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | Ethernet Controller | CAN-FD | SERCOM | TC/Compare | TCC (24-bit/16-bit) | $I^2S$ | USB | QSPI | SDHC | DMA Channels | PCC (data size) | CCL | Position Decoder | RTC | WDT | Frequency Measurement | Event System (Channels) | External Interrupt Lines | I/O Pins | ADC (Channels ADC0/ADC1) | Analog Comparators (Channels) | DAC (Channels) | PTC (Mutual/Self-capacitance Channels) | Temperature Sensor | AES | TRNG | Public Key Cryptography (PUKCC) | Integrity Check Monitor | Tamper Pins |
| SAME53N20 | 1024 | 256 / 8 | 100 | TQFP | Y | N | 8 | 8/2 | 2/3 | Y | Y | Y | 2 | 32 | 14 | 4 | Y | Y | Y | Y | 32 | 16 | 81 | 16/12 | 4 | 2 | 256/32 | Y | Y | Y | Y | Y | 5 |
| SAME53N19 | 512 | 192 / 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME53J20 | 1024 | 256 / 8 | 64 | TQFP, VQFN | | | 6 | 6/2 | | | | | 1 | | 10 | | | | | | | | 51 | 16/8 | | | | | | | | | 3 |
| SAME53J19 | 512 | 192 / 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME53J18 | 256 | 128 / 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME54P20 | 1024 | 256 / 8 | 128 | TQFP | | Y | 8 | 8/2 | | | | | 2 | | 14 | | | | | | | | 99 | 16/16 | | | | | | | | | 5 |
| | | | 120 | TFBGA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME54P19 | 512 | 192 / 8 | 128 | TQFP | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | 120 | TFBGA | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME54N20 | 1024 | 256 / 8 | 100 | TQFP | | | | | | | | | | | | | | | | | | | 81 | 16/12 | | | | | | | | | |
| SAME54N19 | 512 | 192 / 8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

**Table 1-2.** SAM D51/E51 Family Features without Ethernet

| Device | Program Memory (KB) | Data Memory (KB) | Pins | Packages | Peripherals | | | | | | | | | | | | | | | | | | Analog | | | | | Security | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | CAN-FD | SERCOM | TC/Compare | TCC (24-bit/16-bit) | I²S | USB | QSPI | SDHC | DMA Channels | PCC (data size) | CCL | Position Decoder | RTC | WDT | Frequency Measurement | Event System (Channels) | External Interrupt Lines | I/O Pins | ADC (Channels ADC0/ADC1) | Analog Comparators (Channels) | DAC (Channels) | PTC (Mutual/Self-capacitance Channels) | Temperature Sensor | AES | TRNG | Public Key Cryptography (PUKCC) | Integrity Check Monitor | Tamper Pins |
| SAMD51P20 | 1024 | 256/8 | 128 | TQFP | N | 8 | 8/2 | 2/3 | Y | Y | Y | 2 | 32 | 14 | 4 | Y | Y | Y | Y | 32 | 16 | 99 | 16/16 | 4 | 2 | 256/32 | Y | Y | Y | Y | Y | 5 |
| | | | 120 | TFBGA | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMD51P19 | 512 | 192/8 | 128 | TQFP | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | 120 | TFBGA | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMD51N20 | 1024 | 256/8 | 100 | TQFP | | | | | | | | | | | | | | | | | | 81 | 16/12 | | | | | | | | | |
| SAMD51N19 | 512 | 192/8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMD51J20 | 1024 | 256/8 | 64 | TQFP, VQFN, WLCSP | | 6 | 6/2 | | | | | 1 | | 10 | | | | | | | | 51 | 16/8 | | | | | | | | | 3 |
| SAMD51J19 | 512 | 192/8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMD51J18 | 256 | 128/8 | 64 | TQFP, VQFN | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAMD51G19 | 512 | 192/8 | 48 | VQFN | | | 4/2 | 2/1 | N | | | 1 | | 10 | | | | | | | | 37 | 16/4 | | | 121/22 | | | | | | 2 |
| SAMD51G18 | 256 | 128/8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME51N20 | 1024 | 256/8 | 100 | TQFP | Y | 8 | 8/2 | 2/3 | Y | | | 1 | | 14 | | | | | | | | 81 | 16/12 | | | 256/32 | | | | | | 5 |
| SAME51N19 | 512 | 192/8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME51J20 | 1024 | 256/8 | 64 | TQFP, VQFN | | 6 | 6/2 | | | | | 1 | | 10 | | | | | | | | 51 | 16/8 | | | | | | | | | 3 |
| SAME51J19 | 512 | 192/8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME51J18 | 256 | 128/8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| SAME51G18 | 256 | 128/8 | 48 | VQFN | | | 4/2 | 2/1 | N | | | 1 | | 10 | | | | | | | | 37 | 16/4 | | | 121/22 | | | | | | 2 |
| SAME51G19 | 512 | 192/8 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Related Links**

SERCOM I2C Configurations

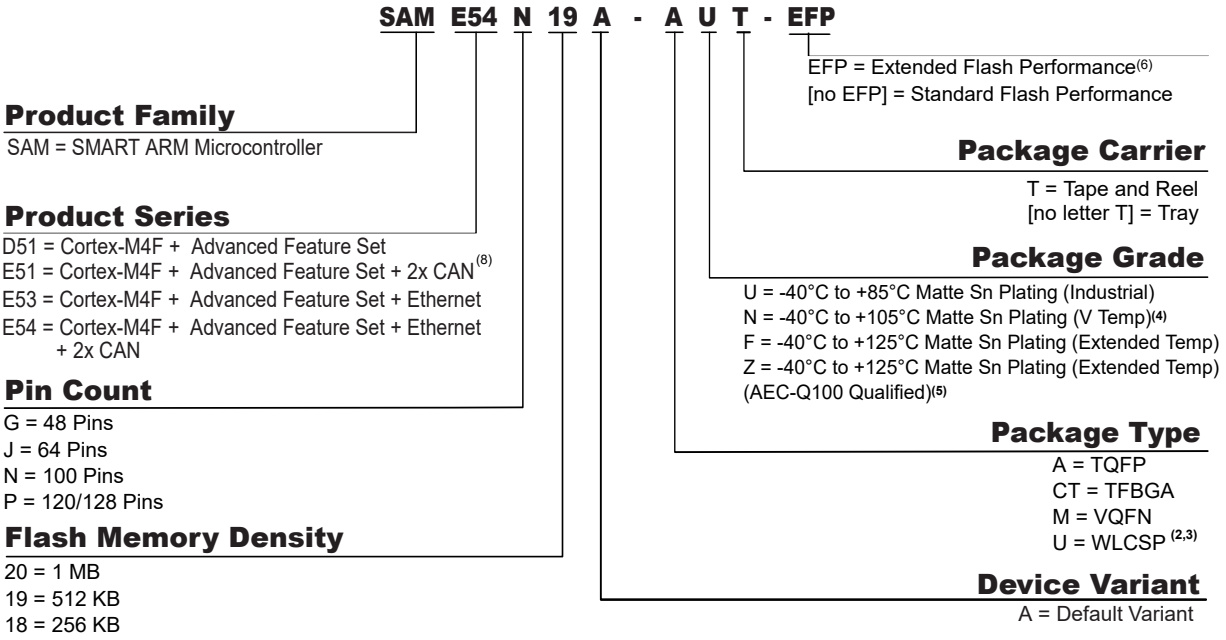
GPIO Clusters


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

# 2. Ordering Information

**Figure 2-1.** Composition of the Ordering Numbers[1]




**Notes:**

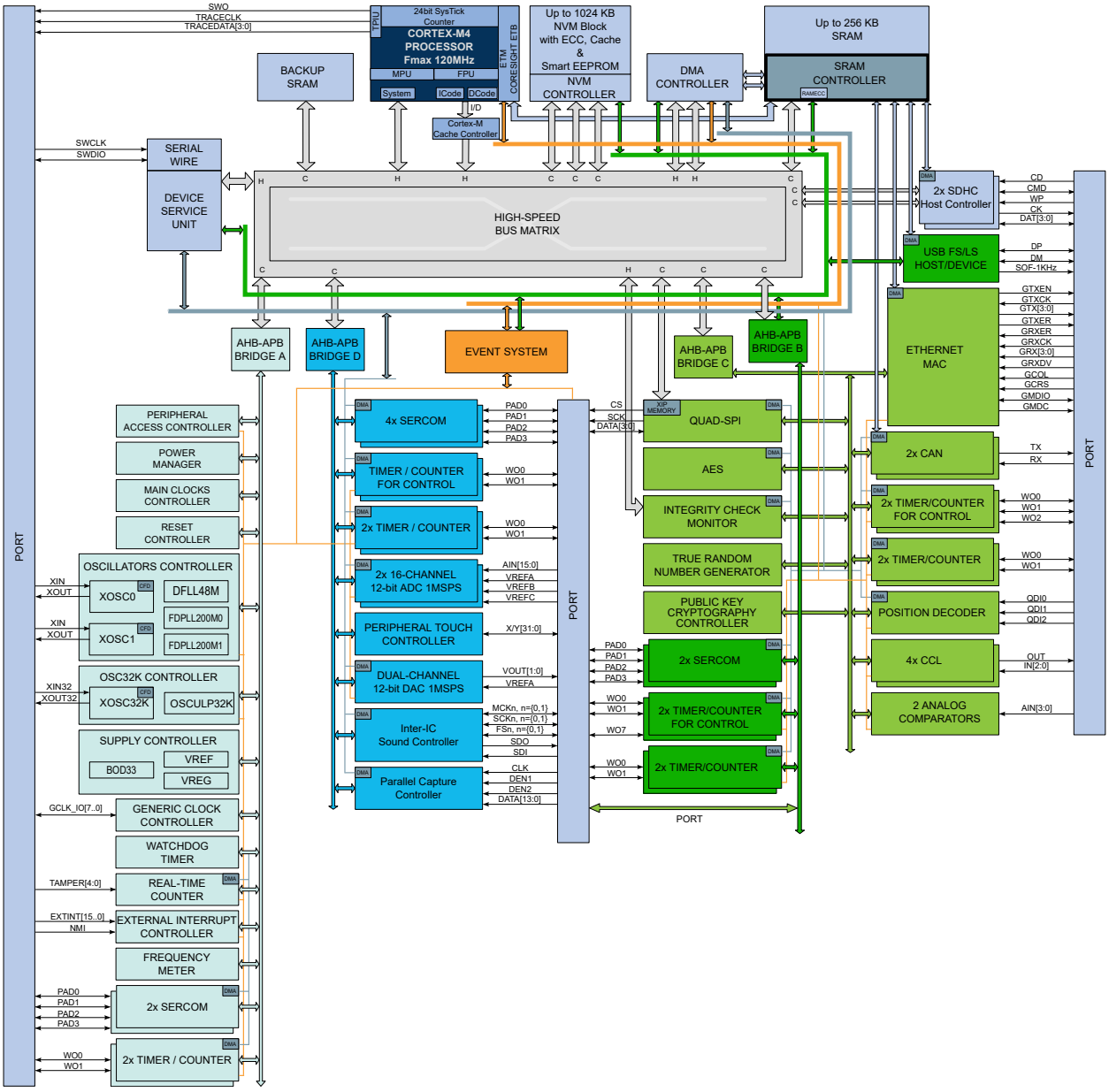
1. Not all combinations are valid. The available device part numbers are listed in Configuration Summary.
2. Devices in the WLCSP package include a factory programmed bootloader. Contact your local Microchip sales office for additional information.
3. The WLCSP package type is available only with the package grade **U** and **N**.
4. Package Grade **N** is available with only package type **U**.
5. The AEC-Q100 Grade 1 qualified version is only offered in the TQFP, VQFN, and BGA packages. The VQFN package will have wettable flanks. The TQFP, VQFN, and BGA packages are assembled with gold bond wires.
6. EFP is only available for package grade **U**.
7. EFP is an ordering code extension and will not be printed onto the package marking (Section 58.1: Package Marking Information).
8. E51G devices have only 1 instance of CAN.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 3. Block Diagram

The actual configuration may vary with device memory and number of pins. Refer to the Configuration Summary for details.

## 3.1. SAM D5x/E5x Block Diagram




**Notes:**

1. Some products have different number of SERCOM instances, Timer/Counter instances, PTC signals, and ADC signals.
2. The block diagram is representing SAM E54P. Refer to the Configuration Summary for the configuration of a given device.

**Related Links**

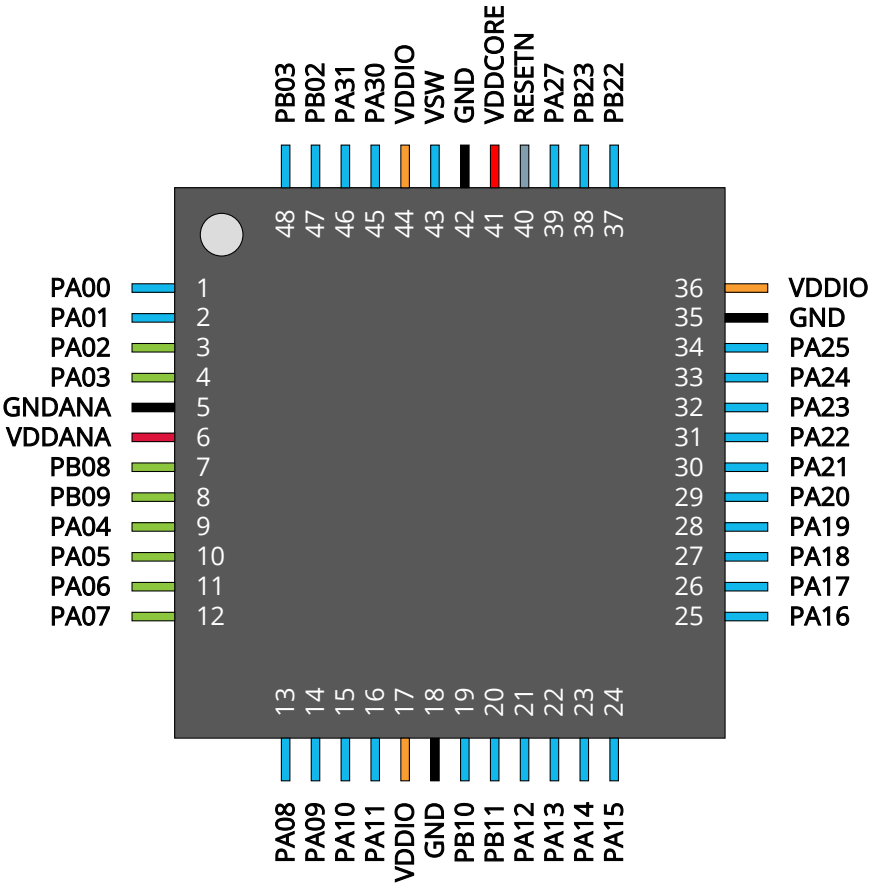
Configuration Summary


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 4. Pinout

## 4.1. 48-Pin VQFN Package

**Figure 4-1.** 48-Pin VQFN(1) Package



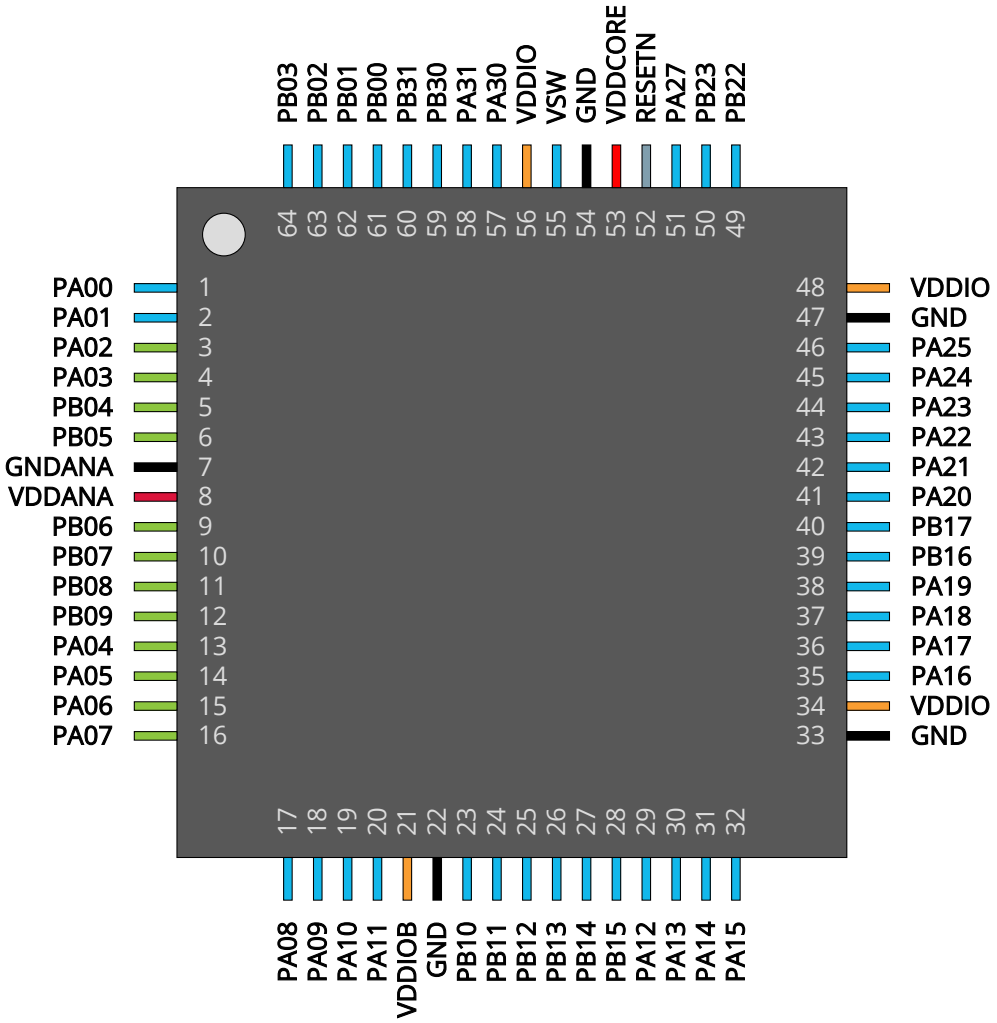

**Note:**

1. It is recommended that the exposed pad be connected to ground in the PCB. Refer to the section Package Drawings for further information.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 4.2. 64-Pin TQFP and VQFN Package

**Figure 4-2.** 64-Pin TQFP and VQFN [1] Package



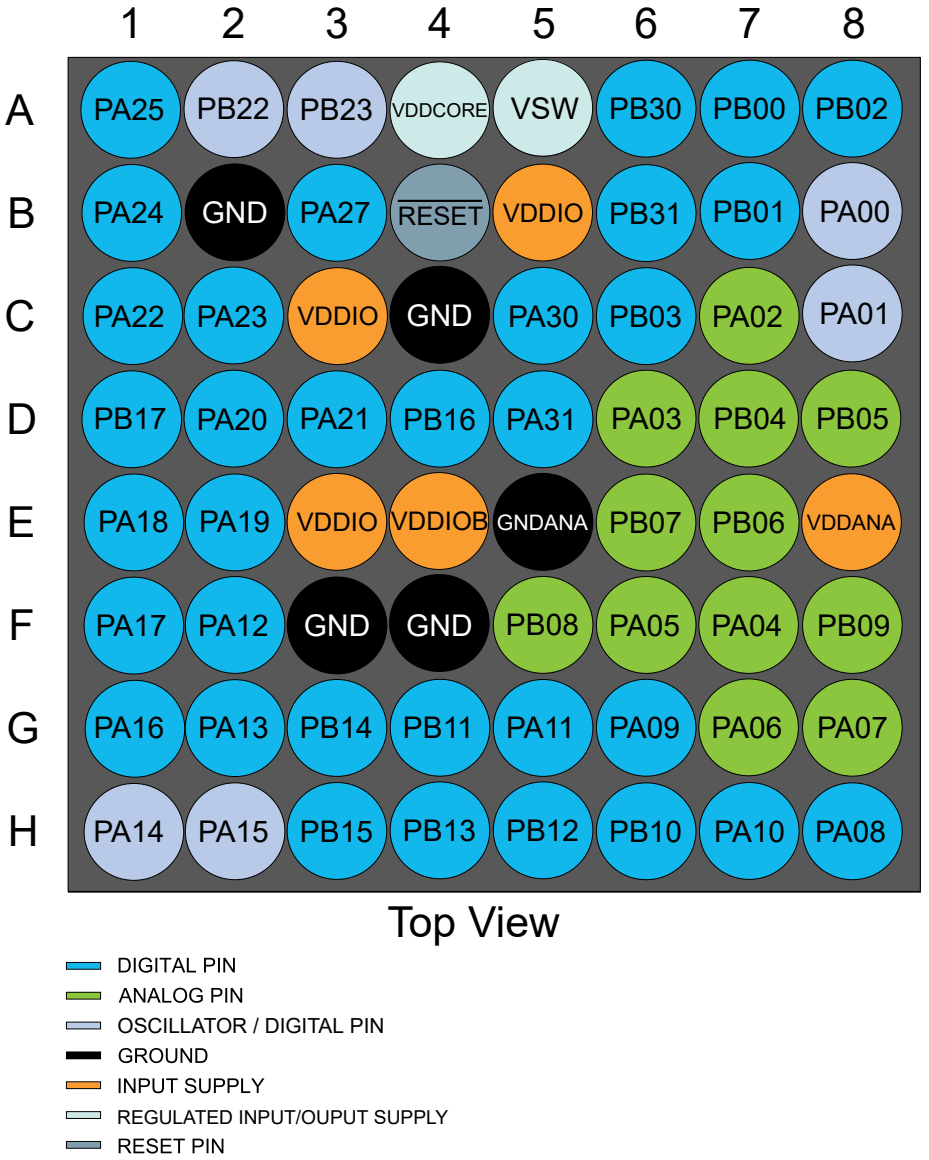

**Note:**

1. It is recommended that the exposed pad be connected to ground in the PCB. Refer to section 58.3 Package Drawings for further information.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 4.3. 64-Pin WLCSP Package

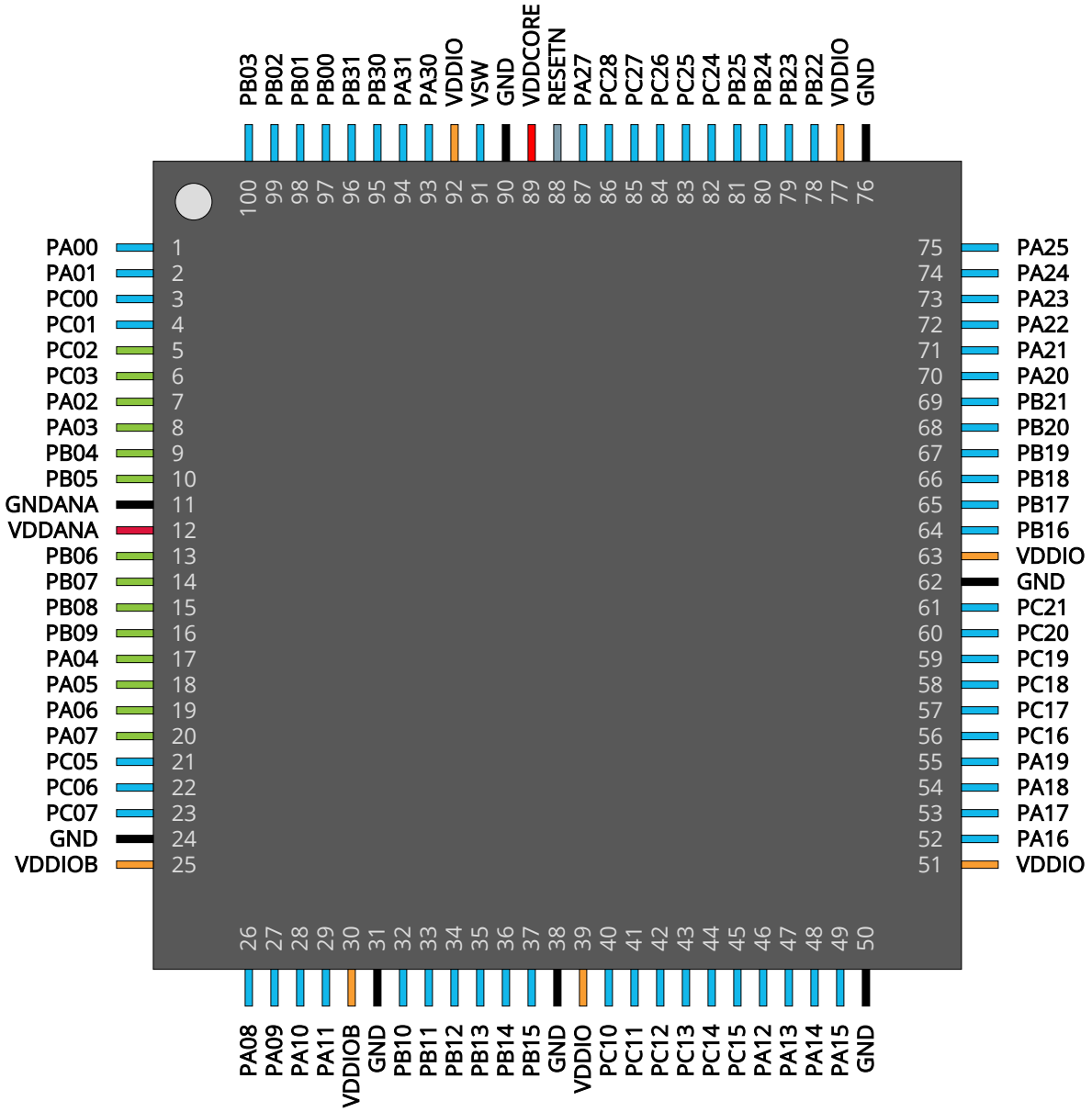




© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 4.4. 100-Pin TQFP Package

**Figure 4-3.** 100-Pin TQFP Package

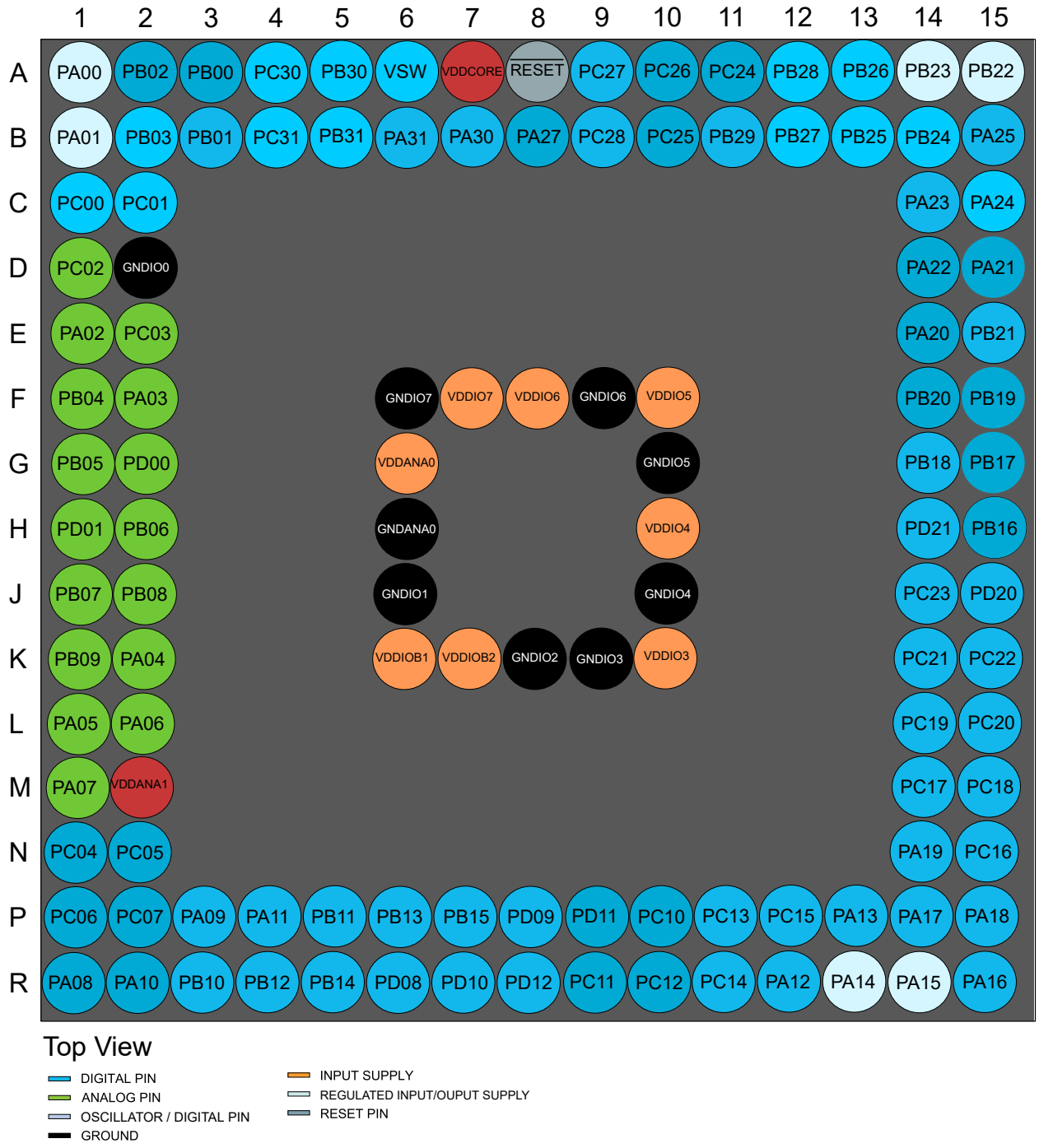




© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 4.5. 120-Ball TFBGA Package

**Figure 4-4.** 120-Ball TFBGA Package

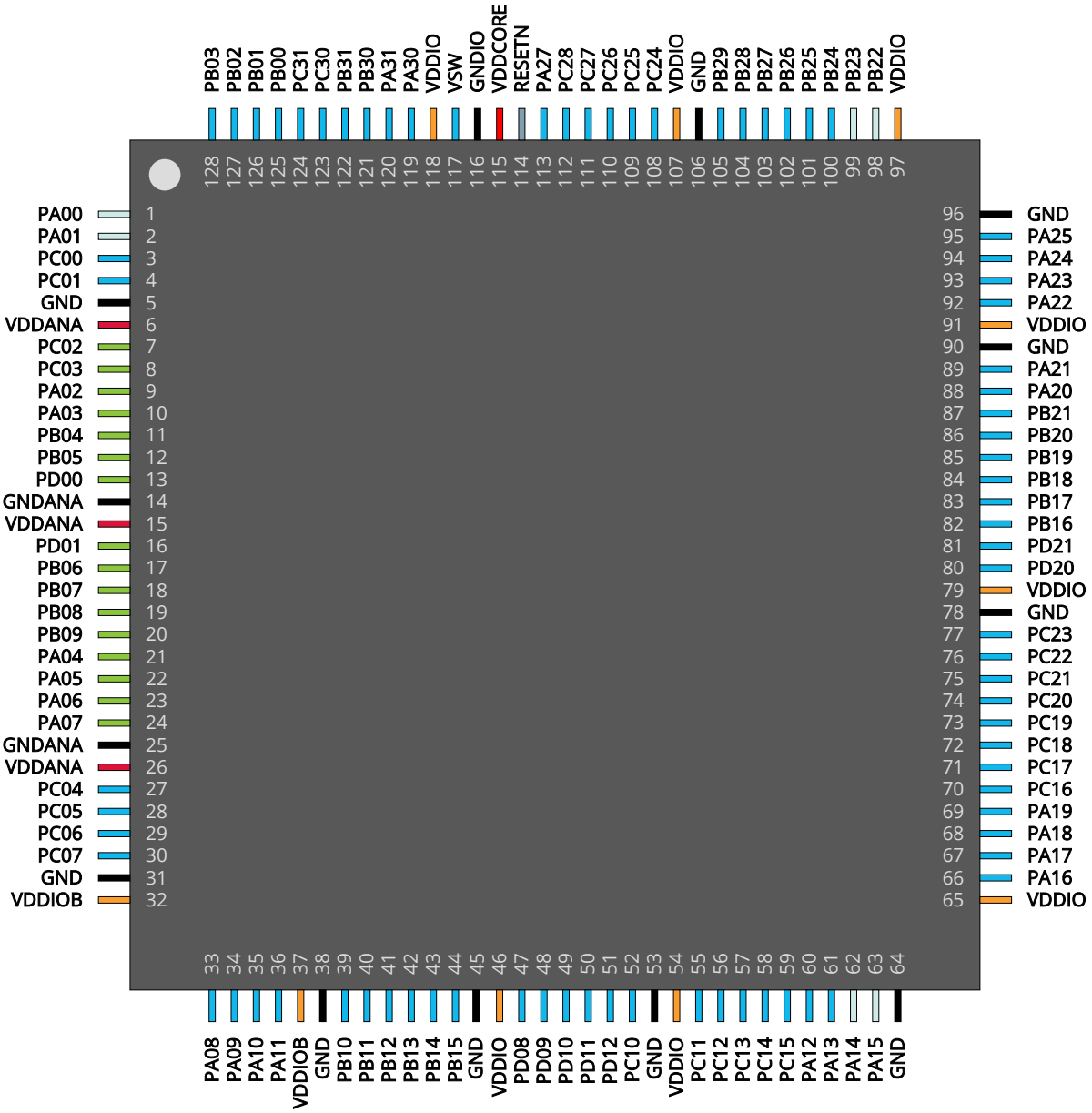




© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 4.6. 128-Pin TQFP Package

**Figure 4-5.** 128-Pin TQFP Package





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 5. Signal Descriptions List

The following table gives details on signal names classified by peripheral.

**Table 5-1.** Signal Descriptions List

| Signal Name | Function | Type | Active Level |
|---|---|---|---|
| **Device Service Unit - DSU** | | | |
| SWCLK | SW Clock | Digital | |
| SWDIO | SW Bidirectional Data | Digital | |
| RESETN | Reset input | Digital | Low |
| **Trace Port Interface Unit - TPIU** | | | |
| TRACEDATA[3:0] | Trace Data Output | Digital | |
| TRACECLK | Trace Clock | Digital | |
| SWO | Serial Wire Output | Digital | |
| **Analog Comparators - AC** | | | |
| CMP[1:0] | AC Comparator Outputs | Digital | |
| AIN[3:0] | AC Analog Inputs | Analog | |
| **Analog Digital Converter - ADC** | | | |
| AIN[15:0] | ADC Analog Inputs | Analog | |
| VREFA | ADC Voltage External Reference A | Analog | |
| VREFB | ADC Voltage External Reference B | Analog | |
| VREFC | ADC Voltage External Reference C | Analog | |
| **Peripheral Touch Controller - PTC** | | | |
| XY[31:0] | PTC X/Y Input/Output | Analog | |
| **Digital Analog Converter - DAC** | | | |
| VOUT[1:0] | DAC Voltage output | Analog | |
| VREFA | DAC Voltage External Reference | Analog | |
| **External Interrupt Controller - EIC** | | | |
| EXTINT[15:0] | External Interrupts inputs | Digital | |
| NMI | External Non-Maskable Interrupt input | Digital | |
| **Generic Clock Generator - GCLK** | | | |
| GCLK_IO[7:0] | Generic Clock (source clock inputs or generic clock generator output) | Digital | |
| **Custom Control Logic - CCL** | | | |
| IN[11:0] | Logic Inputs | Digital | |
| OUT[3:0] | Logic Outputs | Digital | |
| **Supply Controller - SUPC** | | | |
| VBAT | External battery supply Inputs | Analog | |
| OUT[1:0] | Logic Outputs | Digital | |
| **Power Manager - PM** | | | |
| RESETN | Reset input | Digital | Low |
| **Oscillators Control - OSCCTRL** | | | |
| XOSCx - XIN | Crystal or external clock Input | Analog/Digital | |
| XOSCx - XOUT | Crystal Output | Analog | |
| **32KHz Oscillators Control - OSC32KCTRL** | | | |
| XIN32 | 32KHz Crystal or external clock Input | Analog/Digital | |
| XOUT32 | 32KHz Crystal Output | Analog | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 5-1.** Signal Descriptions List (continued)

| Signal Name | Function | Type | Active Level |
|---|---|---|---|
| **General Purpose I/O - PORT** | | | |
| PA31 - PA30, PA27, PA25 - PA00 | Parallel I/O Controller I/O Port A | Digital | |
| PB31 - PB00 | Parallel I/O Controller I/O Port B | Digital | |
| PC31 - PC30, PC28 - PC10, PC07 - PC00 | Parallel I/O Controller I/O Port C | Digital | |
| PD21-PD20, PD12 - PD08, PD01 - PD00 | Parallel I/O Controller I/O Port D | Digital | |
| **Real-Time Counter - RTC** | | | |
| IN[4:0] | Tamper / Wake / Active Layer Protection Input | Digital | |
| OUT | Active Layer Protection Output | Digital | |
| **Timer Counter - TCx** | | | |
| WO[1:0] | Waveform Outputs/Capture Inputs | Digital | |
| **Timer Counter - TCCx** | | | |
| WO[7:0] | Waveform Outputs/Capture Inputs | Digital | |
| **Position Decoder - PDEC** | | | |
| PDEC[2:0] | PDEC Inputs | Digital | |
| **Parallel Capture Controller - PCC** | | | |
| DEN1 | Sensor Sync1 | Digital | |
| DEN2 | Sensor Sync2 | Digital | |
| CLK | Sensor Clock | Digital | |
| DATA[13:0] | Sensor Data | Digital | |
| **Serial Communication Interface - SERCOMx** | | | |
| PAD[3:0] | SERCOM Inputs/Outputs Pads | Digital | |
| **Quad Serial Peripheral Interface - QSPI** | | | |
| SCK | Serial Clock | Digital | |
| CS | Chip Select | Digital | |
| DATA[3:0] | Data Input/Output | Digital | |
| **Ethernet MAC - GMAC** | | | |
| GTXEN | Transmit Enable | Digital | |
| GTXCK | Transmit Clock or Reference Clock | Digital | |
| GTX[3:0] | Transmit Data | Digital | |
| GTXER | Transmit Coding Error | Digital | |
| GRXER | Receive Error | Digital | |
| GRXCK | Receive Clock | Digital | |
| GRX[3:0] | Receive Data | Digital | |
| GRXDV | Receive Data Valid | Digital | |
| GCOL | Collision Detect | Digital | |
| GCRS | Carrier Sense and Data Valid | Digital | |
| GMDIO | Management Data Input/Output | Digital | |
| GMDC | Management Data Clock | Digital | |
| **Universal Serial Bus - USB** | | | |
| DP | DP for USB | Digital | |
| DM | DM for USB | Digital | |
| SOF 1kHz | USB Start of Frame | Digital | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 5-1.** Signal Descriptions List (continued)

| Signal Name | Function | Type | Active Level |
|---|---|---|---|
| **Control Area Network - CANx** | | | |
| TX | CAN Transmit | Digital | |
| RX | CAN Receive | Digital | |
| **Inter-IC Sound Controller - I2S** | | | |
| MCK1, MCK0 | Host Clock | Digital | |
| SCK1, SCK0 | Serial Clock | Digital | |
| FS1, FS0 | I²S Word Select or TDM Frame Sync | Digital | |
| SDO | Serial Data Output for Transmit Serializer | Digital | |
| SDI | Serial Data Input for Receive Serializer | Digital | |
| **SD/MMC Host Controller - SDHCx** | | | |
| CD | SD Card / SDIO / e.MMC Card Detect | Digital | |
| CMD | SD Card / SDIO / e.MMC Command/ Response Line | Digital | |
| WP | SD Card Connector Write Protect Signal | Digital | |
| CK | SD Card / SDIO / e.MMC Clock Signal | Digital | |
| DAT[3:0] | SD Card / SDIO / e.MMC Data Lines | Digital | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 6. I/O Multiplexing and Considerations

## 6.1. Multiplexed Signals

By default, each pin is controlled by the PORT as a general purpose I/O, and alternatively it can be assigned to one of the peripheral functions: A, B, C, D, E, F, G, H, I, J, K, L, M, or N. To enable a peripheral function on a pin, the Peripheral Multiplexer Enable bit in the Pin Configuration register corresponding to that pin (PINCFGn.PMUXEN, n = 0-31) in the PORT must be written to '1'. The selection of peripheral functions, A to N, is done by writing to the Peripheral Multiplexing Odd and Even bits in the Peripheral Multiplexing register (PMUXn.PMUXE/O) of the PORT. The table below describes the peripheral signals multiplexed to the PORT I/O pins.

**Important:** Not all signals are available on all devices. Refer to the Configuration Summary for available peripherals.

**Table 6-1.** Multiplexed Peripheral Signals

| VQFN 48 | TQFP/VQFN/WLCSP 64 | TQFP 100 | TFBGA 120 | TQFP 128 | Pad Name | A | B | | | | | | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | EIC | ANAREF | ADC0 | ADC1 | AC | DAC | PTC | SERCOM | SERCOM | TC | TCC | TCC, PDEC | QSPI, CAN1, USB, CORTEX_CM4 | SDHC, CAN0 | I²S | PCC | GMAC | GCLK, AC | CCL |
| 48 | 64/C6 | 100 | B2 | 128 | PB03 | EIC/EXTINT[3] | - | ADC0/AIN[15] | - | - | - | X21/Y21 | - | SERCOM5/PAD[1] | TC6/WO[1] | - | - | - | - | - | - | - | - | - |
| 1 | 01/B8 | 1 | A1 | 1 | PA00 | EIC/EXTINT[0] | - | - | - | - | - | | - | SERCOM1/PAD[0] | TC2/WO[0] | - | - | - | - | - | - | - | - | - |
| 2 | 02/C8 | 2 | B1 | 2 | PA01 | EIC/EXTINT[1] | - | - | - | - | - | | - | SERCOM1/PAD[1] | TC2/WO[1] | - | - | - | - | - | - | - | - | - |
| | | 3 | C1 | 3 | PC00 | EIC/EXTINT[0] | - | - | ADC1/AIN[10] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| | | 4 | C2 | 4 | PC01 | EIC/EXTINT[1] | - | - | ADC1/AIN[11] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| | | 5 | D1 | 7 | PC02 | EIC/EXTINT[2] | - | - | ADC1/AIN[4] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| | | 6 | E2 | 8 | PC03 | EIC/EXTINT[3] | - | - | ADC1/AIN[5] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| 3 | 03/C7 | 7 | E1 | 9 | PA02 | EIC/EXTINT[2] | - | ADC0/AIN[0] | - | - | DAC/VOUT[0] | | - | - | - | - | - | - | - | - | - | - | - | - |
| 4 | 04/D6 | 8 | F2 | 10 | PA03 | EIC/EXTINT[3] | ANAREF/VREFA | ADC0/AIN[1] | - | - | - | X0/Y0 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 05/D7 | 9 | F1 | 11 | PB04 | EIC/EXTINT[4] | - | - | ADC1/AIN[6] | - | - | X22/Y22 | - | - | - | - | - | - | - | - | - | - | - | - |
| | 06/D8 | 10 | G1 | 12 | PB05 | EIC/EXTINT[5] | - | - | ADC1/AIN[7] | - | - | X23/Y23 | - | - | - | - | - | - | - | - | - | - | - | - |
| | | - | G2 | 13 | PD00 | EIC/EXTINT[0] | - | - | ADC1/AIN[14] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| | | - | H1 | 16 | PD01 | EIC/EXTINT[1] | - | - | ADC1/AIN[15] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| | 09/E7 | 13 | H2 | 17 | PB06 | EIC/EXTINT[6] | - | - | ADC1/AIN[8] | - | - | X24/Y24 | - | - | - | - | - | - | - | - | - | - | - | CCL/IN[6] |
| | 10/E6 | 14 | J1 | 18 | PB07 | EIC/EXTINT[7] | - | - | ADC1/AIN[9] | - | - | X25/Y25 | - | - | - | - | - | - | - | - | - | - | - | CCL/IN[7] |
| 7 | 11/F5 | 15 | J2 | 19 | PB08 | EIC/EXTINT[8] | - | ADC0/AIN[2] | ADC1/AIN[0] | - | - | X1/Y1 | - | SERCOM4/PAD[0] | TC4/WO[0] | - | - | - | - | - | - | - | - | CCL/IN[8] |
| 8 | 12/F8 | 16 | K1 | 20 | PB09 | EIC/EXTINT[9] | - | ADC0/AIN[3] | ADC1/AIN[1] | - | - | X2/Y2 | - | SERCOM4/PAD[1] | TC4/WO[1] | - | - | - | - | - | - | - | - | CCL/OUT[2] |
| 9 | 13/F7 | 17 | K2 | 21 | PA04 | EIC/EXTINT[4] | ANAREF/VREFB | ADC0/AIN[4] | - | AC/AIN[0] | - | X3/Y3 | - | SERCOM0/PAD[0] | TC0/WO[0] | - | - | - | - | - | - | - | - | CCL/IN[0] |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-1.** Multiplexed Peripheral Signals (continued)

| VQFN 48 | TQFP/VQFN/WLCSP 64 | TQFP 100 | TFBGA 120 | TQFP 128 | Pad Name | A | B | | | | | | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | EIC | ANAREF | ADC0 | ADC1 | AC | DAC | PTC | SERCOM | SERCOM | TC | TCC | TCC, PDEC | QSPI, CAN1, USB, CORTEX_CM4 | SDHC, CAN0 | I²S | PCC | GMAC | GCLK, AC | CCL |
| 10 | 14/F6 | 18 | L1 | 22 | PA05 | EIC/EXTINT[5] | - | ADC0/AIN[5] | - | AC/AIN[1] | DAC/VOUT[1] | | - | SERCOM0/PAD[1] | TC0/WO[1] | - | - | - | - | - | - | - | - | CCL/IN[1] |
| 11 | 15/G7 | 19 | L2 | 23 | PA06 | EIC/EXTINT[6] | ANAREF/VREFC | ADC0/AIN[6] | - | AC/AIN[2] | - | X4/Y4 | - | SERCOM0/PAD[2] | TC1/WO[0] | - | - | - | SDHC0/SDCD | - | - | - | - | CCL/IN[2] |
| 12 | 16/G8 | 20 | M1 | 24 | PA07 | EIC/EXTINT[7] | - | ADC0/AIN[7] | - | AC/AIN[3] | - | X5/Y5 | - | SERCOM0/PAD[3] | TC1/WO[1] | - | - | - | SDHC0/SDWP | - | - | - | - | CCL/OUT[0] |
| | | - | N1 | 27 | PC04 | EIC/EXTINT[4] | - | - | - | - | - | | SERCOM6/PAD[0] | - | - | TCC0/WO[0] | - | - | - | - | - | - | - | - |
| | | 21 | N2 | 28 | PC05 | EIC/EXTINT[5] | - | - | - | - | - | | SERCOM6/PAD[1] | - | - | - | - | - | - | - | - | - | - | - |
| | | 22 | P1 | 29 | PC06 | EIC/EXTINT[6] | - | - | - | - | - | | SERCOM6/PAD[2] | - | - | - | - | - | SDHC0/SDCD | - | - | - | - | - |
| | | 23 | P2 | 30 | PC07 | EIC/EXTINT[9] | - | - | - | - | - | | SERCOM6/PAD[3] | - | - | - | - | - | SDHC0/SDWP | - | - | - | - | - |
| 13 | 17/H8 | 26 | R1 | 33 | PA08 | EIC/NMI | - | ADC0/AIN[8] | ADC1/AIN[2] | - | - | X6/Y6 | SERCOM0/PAD[0] | SERCOM2/PAD[1] | TC0/WO[0] | TCC0/WO[0] | TCC1/WO[4] | QSPI/DATA[0] | SDHC0/SDCMD | I²S/MCK[0] | - | - | - | CCL/IN[3] |
| 14 | 18/G6 | 27 | P3 | 34 | PA09 | EIC/EXTINT[9] | - | ADC0/AIN[9] | ADC1/AIN[3] | - | - | X7/Y7 | SERCOM0/PAD[1] | SERCOM2/PAD[0] | TC0/WO[1] | TCC0/WO[1] | TCC1/WO[5] | QSPI/DATA[1] | SDHC0/SDDAT[0] | I²S/FS[0] | - | - | - | CCL/IN[4] |
| 15 | 19/H7 | 28 | R2 | 35 | PA10 | EIC/EXTINT[10] | - | ADC0/AIN[10] | - | - | - | X8/Y8 | SERCOM0/PAD[2] | SERCOM2/PAD[2] | TC1/WO[0] | TCC0/WO[2] | TCC1/WO[6] | QSPI/DATA[2] | SDHC0/SDDAT[1] | I²S/SCK[0] | - | - | GCLK/IO[4] | CCL/IN[5] |
| 16 | 20/G5 | 29 | P4 | 36 | PA11 | EIC/EXTINT[11] | - | ADC0/AIN[11] | - | - | - | X9/Y9 | SERCOM0/PAD[3] | SERCOM2/PAD[3] | TC1/WO[1] | TCC0/WO[3] | TCC1/WO[7] | QSPI/DATA[3] | SDHC0/SDDAT[2] | I²S/SDO | - | - | GCLK/IO[5] | CCL/OUT[1] |
| 19 | 23/H6 | 32 | R3 | 39 | PB10 | EIC/EXTINT[10] | - | - | - | - | - | | - | SERCOM4/PAD[2] | TC5/WO[0] | TCC0/WO[4] | TCC1/WO[0] | QSPI/SCK | SDHC0/SDDAT[3] | I²S/SDI | - | - | GCLK/IO[4] | CCL/IN[11] |
| 20 | 24/G4 | 33 | P5 | 40 | PB11 | EIC/EXTINT[11] | - | - | - | - | - | | - | SERCOM4/PAD[3] | TC5/WO[1] | TCC0/WO[5] | TCC1/WO[1] | QSPI/CS | SDHC0/SDCK | I²S/FS[1] | - | - | GCLK/IO[5] | CCL/OUT[1] |
| | 25/H5 | 34 | R4 | 41 | PB12 | EIC/EXTINT[12] | - | - | - | - | - | X26/Y26 | SERCOM4/PAD[0] | - | TC4/WO[0] | TCC3/WO[0] | TCC0/WO[0] | CAN1/TX | SDHC0/SDCD | I²S/SCK[1] | - | - | GCLK/IO[6] | - |
| | 26/H4 | 35 | P6 | 42 | PB13 | EIC/EXTINT[13] | - | - | - | - | - | X27/Y27 | SERCOM4/PAD[1] | - | TC4/WO[1] | TCC3/WO[1] | TCC0/WO[1] | CAN1/RX | SDHC0/SDWP | I²S/MCK[1] | - | - | GCLK/IO[7] | - |
| | 27/G3 | 36 | R5 | 43 | PB14 | EIC/EXTINT[14] | - | - | - | - | - | X28/Y28 | SERCOM4/PAD[2] | - | TC5/WO[0] | TCC4/WO[0] | TCC0/WO[2] | CAN1/TX | - | - | PCC/DATA[8] | GMAC/GMDC | GCLK/IO[0] | CCL/IN[9] |
| | 28/H3 | 37 | P7 | 44 | PB15 | EIC/EXTINT[15] | - | - | - | - | - | X29/Y29 | SERCOM4/PAD[3] | - | TC5/WO[1] | TCC4/WO[1] | TCC0/WO[3] | CAN1/RX | - | - | PCC/DATA[9] | GMAC/GMDIO | GCLK/IO[1] | CCL/IN[10] |
| | | - | R6 | 47 | PD08 | EIC/EXTINT[3] | - | - | - | - | - | | SERCOM7/PAD[0] | SERCOM6/PAD[1] | - | TCC0/WO[1] | - | - | - | - | - | - | - | - |
| | | - | P8 | 48 | PD09 | EIC/EXTINT[4] | - | - | - | - | - | | SERCOM7/PAD[1] | SERCOM6/PAD[0] | - | TCC0/WO[2] | - | - | - | - | - | - | - | - |
| | | - | R7 | 49 | PD10 | EIC/EXTINT[5] | - | - | - | - | - | | SERCOM7/PAD[2] | SERCOM6/PAD[2] | - | TCC0/WO[3] | - | - | - | - | - | - | - | - |
| | | - | P9 | 50 | PD11 | EIC/EXTINT[6] | - | - | - | - | - | | SERCOM7/PAD[3] | SERCOM6/PAD[3] | - | TCC0/WO[4] | - | - | - | - | - | - | - | - |
| | | - | R8 | 51 | PD12 | EIC/EXTINT[7] | - | - | - | - | - | | - | - | - | TCC0/WO[5] | - | - | - | - | - | - | - | - |
| | | 40 | P10 | 52 | PC10 | EIC/EXTINT[10] | - | - | - | - | - | | SERCOM6/PAD[2] | SERCOM7/PAD[2] | - | TCC0/WO[0] | TCC1/WO[4] | - | - | - | - | - | - | - |
| | | 41 | R9 | 55 | PC11 | EIC/EXTINT[11] | - | - | - | - | - | | SERCOM6/PAD[3] | SERCOM7/PAD[3] | - | TCC0/WO[1] | TCC1/WO[5] | - | - | - | - | GMAC/GMDC | - | - |
| | | 42 | R10 | 56 | PC12 | EIC/EXTINT[12] | - | - | - | - | - | | SERCOM7/PAD[0] | SERCOM6/PAD[1] | - | TCC0/WO[2] | TCC1/WO[6] | - | - | - | PCC/DATA[10] | GMAC/GMDIO | - | - |
| | | 43 | P11 | 57 | PC13 | EIC/EXTINT[13] | - | - | - | - | - | | SERCOM7/PAD[1] | SERCOM6/PAD[0] | - | TCC0/WO[3] | TCC1/WO[7] | - | - | - | PCC/DATA[11] | - | - | - |
| | | 44 | R11 | 58 | PC14 | EIC/EXTINT[14] | - | - | - | - | - | | SERCOM7/PAD[2] | SERCOM6/PAD[2] | - | TCC0/WO[4] | TCC1/WO[0] | - | - | - | PCC/DATA[12] | GMAC/GRX[3] | - | - |
| | | 45 | P12 | 59 | PC15 | EIC/EXTINT[15] | - | - | - | - | - | | SERCOM7/PAD[3] | SERCOM6/PAD[3] | - | TCC0/WO[5] | TCC1/WO[1] | - | - | - | PCC/DATA[13] | GMAC/GRX[2] | - | - |


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

**Table 6-1.** Multiplexed Peripheral Signals (continued)

| VQFN 48 | TQFP/VQFN/WLCSP 64 | TQFP 100 | TFBGA 120 | TQFP 128 | Pad Name | A | B | | | | | | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | EIC | ANAREF | ADC0 | ADC1 | AC | DAC | PTC | SERCOM | SERCOM | TC | TCC | TCC, PDEC | QSPI, CAN1, USB, CORTEX_CM4 | SDHC, CAN0 | I²S | PCC | GMAC | GCLK, AC | CCL |
| 21 | 29/F2 | 46 | R12 | 60 | PA12 | EIC/EXTINT[12] | - | - | - | - | - | | SERCOM2/PAD[0] | SERCOM4/PAD[1] | TC2/WO[0] | TCC0/WO[6] | TCC1/WO[2] | - | SDHC0/SDCD | - | PCC/DEN1 | GMAC/GRX[1] | AC/CMP[0] | - |
| 22 | 30/G2 | 47 | P13 | 61 | PA13 | EIC/EXTINT[13] | - | - | - | - | - | | SERCOM2/PAD[1] | SERCOM4/PAD[0] | TC2/WO[1] | TCC0/WO[7] | TCC1/WO[3] | - | SDHC0/SDWP | - | PCC/DEN2 | GMAC/GRX[0] | AC/CMP[1] | - |
| 23 | 31/H1 | 48 | R13 | 62 | PA14 | EIC/EXTINT[14] | - | - | - | - | - | | SERCOM2/PAD[2] | SERCOM4/PAD[2] | TC3/WO[0] | TCC2/WO[0] | TCC1/WO[2] | - | - | - | PCC/CLK | GMAC/GTXCK | GCLK/IO[0] | - |
| 24 | 32/H2 | 49 | R14 | 63 | PA15 | EIC/EXTINT[15] | - | - | - | - | - | | SERCOM2/PAD[3] | SERCOM4/PAD[3] | TC3/WO[1] | TCC2/WO[1] | TCC1/WO[3] | - | - | - | - | GMAC/GRXER | GCLK/IO[1] | - |
| 25 | 35/G1 | 52 | R15 | 66 | PA16 | EIC/EXTINT[0] | - | - | - | - | - | X10/Y10 | SERCOM1/PAD[0] | SERCOM3/PAD[1] | TC2/WO[0] | TCC1/WO[0] | TCC0/WO[4] | - | - | - | PCC/DATA[0] | GMAC/GCRS/GRXDV[6] | GCLK/IO[2] | CCL/IN[0] |
| 26 | 36/F1 | 53 | P14 | 67 | PA17 | EIC/EXTINT[1] | - | - | - | - | - | X11/Y11 | SERCOM1/PAD[1] | SERCOM3/PAD[0] | TC2/WO[1] | TCC1/WO[1] | TCC0/WO[5] | - | - | - | PCC/DATA[1] | GMAC/GTXEN | GCLK/IO[3] | CCL/IN[1] |
| 27 | 37/E1 | 54 | P15 | 68 | PA18 | EIC/EXTINT[2] | - | - | - | - | - | X12/Y12 | SERCOM1/PAD[2] | SERCOM3/PAD[2] | TC3/WO[0] | TCC1/WO[2] | TCC0/WO[6] | - | - | - | PCC/DATA[2] | GMAC/GTX[0] | AC/CMP[0] | CCL/IN[2] |
| 28 | 38/E2 | 55 | N14 | 69 | PA19 | EIC/EXTINT[3] | - | - | - | - | - | X13/Y13 | SERCOM1/PAD[3] | SERCOM3/PAD[3] | TC3/WO[1] | TCC1/WO[3] | TCC0/WO[7] | - | - | - | PCC/DATA[3] | GMAC/GTX[1] | AC/CMP[1] | CCL/OUT[0] |
| | | 56 | N15 | 70 | PC16 | EIC/EXTINT[0] | - | - | - | - | - | | SERCOM6/PAD[0] | SERCOM0/PAD[1] | - | TCC0/WO[0] | PDEC/PDEC[0] | - | - | - | - | GMAC/GTX[2] | - | - |
| | | 57 | M14 | 71 | PC17 | EIC/EXTINT[1] | - | - | - | - | - | | SERCOM6/PAD[1] | SERCOM0/PAD[0] | - | TCC0/WO[1] | PDEC/PDEC[1] | - | - | - | - | GMAC/GTX[3] | - | - |
| | | 58 | M15 | 72 | PC18 | EIC/EXTINT[2] | - | - | - | - | - | | SERCOM6/PAD[2] | SERCOM0/PAD[2] | - | TCC0/WO[2] | PDEC/PDEC[2] | - | - | - | - | GMAC/GRXCK | - | - |
| | | 59 | L14 | 73 | PC19 | EIC/EXTINT[3] | - | - | - | - | - | | SERCOM6/PAD[3] | SERCOM0/PAD[3] | - | TCC0/WO[3] | - | - | - | - | - | GMAC/GTXER | - | - |
| | | 60 | L15 | 74 | PC20 | EIC/EXTINT[4] | - | - | - | - | - | | - | - | - | TCC0/WO[4] | - | - | SDHC1/SDCD | - | - | GMAC/GRXDV | - | CCL/IN[9] |
| | | 61 | K14 | 75 | PC21 | EIC/EXTINT[5] | - | - | - | - | - | | - | - | - | TCC0/WO[5] | - | - | SDHC1/SDWP | - | - | GMAC/GCOL | - | CCL/IN[10] |
| | | - | K15 | 76 | PC22 | EIC/EXTINT[6] | - | - | - | - | - | | SERCOM1/PAD[0] | SERCOM3/PAD[1] | - | TCC0/WO[6] | - | - | - | - | - | GMAC/GMDC | - | - |
| | | - | J14 | 77 | PC23 | EIC/EXTINT[7] | - | - | - | - | - | | SERCOM1/PAD[1] | SERCOM3/PAD[0] | - | TCC0/WO[7] | - | - | - | - | - | GMAC/GMDIO | - | - |
| | | - | J15 | 80 | PD20 | EIC/EXTINT[10] | - | - | - | - | - | | SERCOM1/PAD[2] | SERCOM3/PAD[2] | - | TCC1/WO[0] | - | - | SDHC1/SDCD | - | - | - | - | - |
| | | - | H14 | 81 | PD21 | EIC/EXTINT[11] | - | - | - | - | - | | SERCOM1/PAD[3] | SERCOM3/PAD[3] | - | TCC1/WO[1] | - | - | SDHC1/SDWP | - | - | - | - | - |
| | 39/D4 | 64 | H15 | 82 | PB16 | EIC/EXTINT[0] | - | - | - | - | - | | SERCOM5/PAD[0] | - | TC6/WO[0] | TCC3/WO[0] | TCC0/WO[4] | - | SDHC1/SDCD | I²S/SCK[0] | - | - | GCLK/IO[2] | CCL/IN[11] |
| | 40/D1 | 65 | G15 | 83 | PB17 | EIC/EXTINT[1] | - | - | - | - | - | | SERCOM5/PAD[1] | - | TC6/WO[1] | TCC3/WO[1] | TCC0/WO[5] | - | SDHC1/SDWP | I²S/MCK[0] | - | - | GCLK/IO[3] | CCL/OUT[3] |
| | | 66 | G14 | 84 | PB18 | EIC/EXTINT[2] | - | - | - | - | - | | SERCOM5/PAD[2] | SERCOM7/PAD[2] | - | TCC1/WO[0] | PDEC/PDEC[0] | - | SDHC1/SDDAT[0] | - | - | - | GCLK/IO[4] | - |
| | | 67 | F15 | 85 | PB19 | EIC/EXTINT[3] | - | - | - | - | - | | SERCOM5/PAD[3] | SERCOM7/PAD[3] | - | TCC1/WO[1] | PDEC/PDEC[1] | - | SDHC1/SDDAT[1] | - | - | - | GCLK/IO[5] | - |
| | | 68 | F14 | 86 | PB20 | EIC/EXTINT[4] | - | - | - | - | - | | SERCOM3/PAD[0] | SERCOM7/PAD[1] | - | TCC1/WO[2] | PDEC/PDEC[2] | - | SDHC1/SDDAT[2] | - | - | - | GCLK/IO[6] | - |
| | | 69 | E15 | 87 | PB21 | EIC/EXTINT[5] | - | - | - | - | - | | SERCOM3/PAD[1] | SERCOM7/PAD[0] | - | TCC1/WO[3] | - | - | SDHC1/SDDAT[3] | - | - | - | GCLK/IO[7] | - |
| 29 | 41/D2 | 70 | E14 | 88 | PA20 | EIC/EXTINT[4] | - | - | - | - | - | X14/Y14 | SERCOM5/PAD[2] | SERCOM3/PAD[2] | TC7/WO[0] | TCC1/WO[4] | TCC0/WO[0] | - | SDHC1/SDCMD | I²S/FS[0] | PCC/DATA[4] | GMAC/GMDC | - | - |
| 30 | 42/D3 | 71 | D15 | 89 | PA21 | EIC/EXTINT[5] | - | - | - | - | - | X15/Y15 | SERCOM5/PAD[3] | SERCOM3/PAD[3] | TC7/WO[1] | TCC1/WO[5] | TCC0/WO[1] | - | SDHC1/SDCK | I²S/SDO | PCC/DATA[5] | GMAC/GMDIO | - | - |
| 31 | 43/C1 | 72 | D14 | 92 | PA22 | EIC/EXTINT[6] | - | - | - | - | - | X16/Y16 | SERCOM3/PAD[0] | SERCOM5/PAD[1] | TC4/WO[0] | TCC1/WO[6] | TCC0/WO[2] | - | CAN0/TX | I²S/SDI | PCC/DATA[6] | - | - | CCL/IN[6] |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-1.** Multiplexed Peripheral Signals (continued)

| VQFN 48 | TQFP/VQFN/WLCSP 64 | TQFP 100 | TFBGA 120 | TQFP 128 | Pad Name | A | B | | | | | | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | EIC | ANAREF | ADC0 | ADC1 | AC | DAC | PTC | SERCOM | SERCOM | TC | TCC | TCC, PDEC | QSPI, CAN1, USB, CORTEX_CM4 | SDHC, CAN0 | I²S | PCC | GMAC | GCLK, AC | CCL |
| 32 | 44/C2 | 73 | C14 | 93 | PA23 | EIC/EXTINT[7] | - | - | - | - | - | X17/Y17 | SERCOM3/PAD[1] | SERCOM5/PAD[0] | TC4/WO[1] | TCC1/WO[7] | TCC0/WO[3] | USB/SOF_1KHZ | CAN0/RX | I²S/FS[1] | PCC/DATA[7] | - | - | CCL/IN[7] |
| 33 | 45/B1 | 74 | C15 | 94 | PA24 | EIC/EXTINT[8] | - | - | - | - | - | | SERCOM3/PAD[2] | SERCOM5/PAD[2] | TC5/WO[0] | TCC2/WO[2] | PDEC/PDEC[0] | USB/DM | CAN0/TX | - | - | - | - | CCL/IN[8] |
| 34 | 46/A1 | 75 | B15 | 95 | PA25 | EIC/EXTINT[9] | - | - | - | - | - | | SERCOM3/PAD[3] | SERCOM5/PAD[3] | TC5/WO[1] | - | PDEC/PDEC[1] | USB/DP | CAN0/RX | - | - | - | - | CCL/OUT[2] |
| 37 | 49/A2 | 78 | A15 | 98 | PB22 | EIC/EXTINT[6] | - | - | - | - | - | | SERCOM1/PAD[2] | SERCOM5/PAD[2] | TC7/WO[0] | - | PDEC/PDEC[2] | USB/SOF_1KHZ | - | - | - | - | GCLK/IO[0] | CCL/IN[0] |
| 38 | 50/A3 | 79 | A14 | 99 | PB23 | EIC/EXTINT[7] | - | - | - | - | - | | SERCOM1/PAD[3] | SERCOM5/PAD[3] | TC7/WO[1] | - | PDEC/PDEC[0] | - | - | - | - | - | GCLK/IO[1] | CCL/OUT[0] |
| | | 80 | B14 | 100 | PB24 | EIC/EXTINT[8] | - | - | - | - | - | | SERCOM0/PAD[0] | SERCOM2/PAD[1] | - | - | PDEC/PDEC[1] | - | - | - | - | - | AC/CMP[0] | - |
| | | 81 | B13 | 101 | PB25 | EIC/EXTINT[9] | - | - | - | - | - | | SERCOM0/PAD[1] | SERCOM2/PAD[0] | - | - | PDEC/PDEC[2] | - | - | - | - | - | AC/CMP[1] | - |
| | | - | A13 | 102 | PB26 | EIC/EXTINT[12] | - | - | - | - | - | | SERCOM2/PAD[0] | SERCOM4/PAD[1] | - | TCC1/WO[2] | - | - | - | - | - | - | - | - |
| | | - | B12 | 103 | PB27 | EIC/EXTINT[13] | - | - | - | - | - | | SERCOM2/PAD[1] | SERCOM4/PAD[0] | - | TCC1/WO[3] | - | - | - | - | - | - | - | - |
| | | - | A12 | 104 | PB28 | EIC/EXTINT[14] | - | - | - | - | - | | SERCOM2/PAD[2] | SERCOM4/PAD[2] | - | TCC1/WO[4] | - | - | - | I²S/SCK[1] | - | - | - | - |
| | | - | B11 | 105 | PB29 | EIC/EXTINT[15] | - | - | - | - | - | | SERCOM2/PAD[3] | SERCOM4/PAD[3] | - | TCC1/WO[5] | - | - | - | I²S/MCK[1] | - | - | - | - |
| | | 82 | A11 | 108 | PC24 | EIC/EXTINT[8] | - | - | - | - | - | | SERCOM0/PAD[2] | SERCOM2/PAD[2] | - | - | - | CORTEX_CM4/TRACEDATA[3] | - | - | - | - | - | - |
| | | 83 | B10 | 109 | PC25 | EIC/EXTINT[9] | - | - | - | - | - | | SERCOM0/PAD[3] | SERCOM2/PAD[3] | - | - | - | CORTEX_CM4/TRACEDATA[2] | - | - | - | - | - | - |
| | | 84 | A10 | 110 | PC26 | EIC/EXTINT[10] | - | - | - | - | - | | - | - | - | - | - | CORTEX_CM4/TRACEDATA[1] | - | - | - | - | - | - |
| | | 85 | A9 | 111 | PC27 | EIC/EXTINT[11] | - | - | - | - | - | | SERCOM1/PAD[0] | - | - | - | - | CORTEX_CM4/TRACECLK | - | - | - | - | CORTEX_M4/SWO | CCL/IN[4] |
| | | 86 | B9 | 112 | PC28 | EIC/EXTINT[12] | - | - | - | - | - | | SERCOM1/PAD[1] | - | - | - | - | CORTEX_CM4/TRACEDATA[0] | - | - | - | - | - | CCL/IN[5] |
| 39 | 51/B3 | 87 | B8 | 113 | PA27 | EIC/EXTINT[11] | - | - | - | - | - | X18/Y18 | - | - | - | - | - | | - | - | - | - | GCLK/IO[1] | - |
| 40 | 52/B4 | 88 | A8 | 114 | RESET_N | - | - | - | - | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| 45 | 57/C5 | 93 | B7 | 119 | PA30 | EIC/EXTINT[14] | - | - | - | - | - | X19/Y19 | SERCOM7/PAD[2] | SERCOM1/PAD[2] | TC6/WO[0] | TCC2/WO[0] | - | CORTEX_CM4/SWCLK | - | - | - | - | GCLK/IO[0] | CCL/IN[3] |
| 46 | 58/D5 | 94 | B6 | 120 | PA31 | EIC/EXTINT[15] | - | - | - | - | - | | SERCOM7/PAD[3] | SERCOM1/PAD[3] | TC6/WO[1] | TCC2/WO[1] | - | CORTEX_CM4/SWDIO | - | - | - | - | - | CCL/OUT[1] |
| | 59/A6 | 95 | A5 | 121 | PB30 | EIC/EXTINT[14] | - | - | - | - | - | | SERCOM7/PAD[0] | SERCOM5/PAD[1] | TC0/WO[0] | TCC4/WO[0] | TCC0/WO[6] | CORTEX_CM4/SWO | - | - | - | - | - | - |
| | 60/B6 | 96 | B5 | 122 | PB31 | EIC/EXTINT[15] | - | - | - | - | - | | SERCOM7/PAD[1] | SERCOM5/PAD[0] | TC0/WO[1] | TCC4/WO[1] | TCC0/WO[7] | | - | - | - | - | - | - |
| | | - | A4 | 123 | PC30 | EIC/EXTINT[14] | - | - | ADC1/AIN[12] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| | | - | B4 | 124 | PC31 | EIC/EXTINT[15] | - | - | ADC1/AIN[13] | - | - | | - | - | - | - | - | - | - | - | - | - | - | - |
| | 61/A7 | 97 | A3 | 125 | PB00 | EIC/EXTINT[0] | - | ADC0/AIN[12] | - | - | - | X30/Y30 | - | SERCOM5/PAD[2] | TC7/WO[0] | - | - | - | - | - | - | - | - | CCL/IN[1] |
| | 62/B7 | 98 | B3 | 126 | PB01 | EIC/EXTINT[1] | - | ADC0/AIN[13] | - | - | - | X31/Y31 | - | SERCOM5/PAD[3] | TC7/WO[1] | - | - | - | - | - | - | - | - | CCL/IN[2] |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-1.** Multiplexed Peripheral Signals (continued)

| VQFN 48 | TQFP/VQFN/WLCSP 64 | TQFP 100 | TFBGA 120 | TQFP 128 | Pad Name | A | B | | | | | | C | D | E | F | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | EIC | ANAREF | ADC0 | ADC1 | AC | DAC | PTC | SERCOM | SERCOM | TC | TCC | TCC, PDEC | QSPI, CAN1, USB, CORTEX_CM4 | SDHC, CAN0 | I²S | PCC | GMAC | GCLK, AC | CCL |
| 47 | 63/A8 | 99 | A2 | 127 | PB02 | EIC/EXTINT[2] | - | ADC0/AIN[14] | - | - | - | X20/Y20 | - | SERCOM5/PAD[0] | TC6/WO[0] | TCC2/WO[2] | - | - | - | - | - | - | - | CCL/OUT[0] |

**Notes:**

1. All analog pin functions are on the peripheral function B. The peripheral function B must be selected to disable the digital control of the pin. The AC has analog signals on the peripheral function B and digital signals on the peripheral function M.
2. The pins used by the SERCOM in I²C mode are listed in the section *SERCOM I²C Configurations.*
3. The following High Sink pins have different properties than the regular pins: PA08, PA09, PA12, PA13, PA16, PA17, PA22, PA23, PD08, and PD09.
4. Clusters of multiple GPIO pins are sharing the same supply pin.
5. When TRACE is used in single-wire debug mode, PC27 assumes the role of SWO. In other debug modes, PB30 assumes the SWO functionality.
6. GRXDV is available on PA16 for the 64-pin package only.

**Important:** Not all signals are available on all devices. Refer to the Configuration Summary for available peripherals.

**Related Links**

SERCOM I2C Configurations

GPIO Clusters

## 6.2. Other Functions

### 6.2.1. Oscillator Pinout

The oscillators are not mapped to the normal PORT functions and their multiplexing is controlled by registers in the Oscillators Controller (OSCCTRL) and in the 32K Oscillators Controller (OSC32KCTRL).

**Table 6-2.** Oscillator Pinout

| Oscillator | Supply | Signal | I/O pin |
|---|---|---|---|
| XOSC0 | VDDIO | XIN | PA14 |
| | | XOUT | PA15 |
| XOSC1 | VDDIO | XIN | PB22 |
| | | XOUT | PB23 |
| XOSC32K | VSWOUT | XIN32 | PA00 |
| | | XOUT32 | PA01 |

**Note:** To guarantee the XOSC32K behavior in crystal mode, PC00 must be static.

**Table 6-3.** XOSC32K Jitter Minimization

| Package Pin Count | Steady Signal Recommended |
|---|---|
| 128 | PB00, PB01, PB02, PB03, PC00, PC01 |
| 100 | PB00, PB01, PB02, PB03, PC00, PC01 |
| 120 | PB00, PB01, PB02, PB03, PC00, PC01 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-3.** XOSC32K Jitter Minimization (continued)

| Package Pin Count | Steady Signal Recommended |
|---|---|
| 64 | PB00,PB01,PB02,PB03, PA02,PA03 |
| 48 | PB02, PB03,PA02,PA03 |

### 6.2.2. Serial Wire Debug Interface Pinout

Only the SWCLK pin is mapped to the normal PORT functions. A debugger cold-plugging or hot-plugging detection will automatically switch the SWDIO port to the SWDIO function.

**Table 6-4.** Serial Wire Debug Interface Pinout

| Signal | Supply | I/O pin |
|---|---|---|
| SWCLK | VDDIO | PA30 |
| SWDIO | VDDIO | PA31 |

### 6.2.3. Trace Port Interface Unit Pinout

The Embedded Trace Module (ETM) is leaning on Trace Port Interface Unit (TPIU) to export data out of the system.

**Table 6-5.** Trace Port Interface Unit Pinout

| Signal | Supply | I/O pin |
|---|---|---|
| TRACE DATA[3] | VDDIO | PC24 |
| TRACE DATA[2] | VDDIO | PC25 |
| TRACE DATA[1] | VDDIO | PC26 |
| TRACE DATA[0] | VDDIO | PC28 |
| TRACE CLK | VDDIO | PC27 |
| SWO | VDDIO | PB30, PC27 |

### 6.2.4. Supply Controller Pinout

The outputs of the Supply Controller (SUPC) are not mapped to the normal PORT functions. They are controlled by registers in the SUPC.

**Table 6-6.** SUPC Pinout

| Signal | I/O pin |
|---|---|
| OUT0 | PB01 |
| OUT1 | PB02 |

**Note:** If the RTC is enabled to use the pins shared with the SUPC, the RTC will have higher priority.

### 6.2.5. RTC Pinout

The pins used for Tamper Detection by the Real Time Counter (RTC) are not mapped to the regular PORT functions. These pins and their multiplexing is controlled by register settings of the RTC. If many pins of the tamper detection feature is not used by the RTC, then the pin could be used for other I/O functions, by ensuring the corresponding TAMPCTRL.INACT function is disabled.

**Table 6-7.** RTC Pinout

| RTC Signal | I/O Pin |
|---|---|
| IN0 | PB00 |
| IN1 | PB02 |
| IN2 | PA02 |
| IN3 | PC00 |
| IN4 | PC01 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-7.** RTC Pinout (continued)

| RTC Signal | I/O Pin |
|---|---|
| OUT | PB01 |

**Important:** If Supply Controller (SUPC) and RTC are configured to drive pin PB1 or pin PB2, then the RTC has priority.

### 6.2.6. SERCOM I²C Configurations

The SAM D5x/E5x has up to eight instances of the serial communication interface (SERCOM) peripheral. All instances support USART, including RS485 and ISO7816, SPI and I²C protocols. The following table lists the I²C pins location.

**Table 6-8.** SERCOM I²C Pinout

| Package Pin Count | Supply | I/O pins with I²C Support |
|---|---|---|
| 128 | VDDIOB | PA08, PA09 |
| | VDDIO | PA12, PA13, PA16, PA17, PA22, PA23, PD08, PD09 |
| 120 | VDDIOB | PA08, PA09 |
| | VDDIO | PA12, PA13, PA16, PA17, PA22, PA23, PD08, PD09 |
| 100 | VDDIOB | PA08, PA09 |
| | VDDIO | PA12, PA13, PA16, PA17, PA22, PA23 |
| 64 | VDDIOB | PA08, PA09 |
| | VDDIO | PA12, PA13, PA16, PA17, PA22, PA23 |
| 48 | VDDIO | PA08, PA09, PA12, PA13, PA16, PA17, PA22, PA23 |

### 6.2.7. TCC Configurations

The SAM D5x/E5x has five instances of the Timer/Counter for Control applications (TCC) peripheral, TCC[4:0]. The following table lists the features for each TCC instance.

**Table 6-9.** TCC Configuration Summary

| TCC# | Channels (CC_NUM) | Waveform Output (WO_NUM) | Counter size | Fault | Dithering | Output matrix | Dead Time Insertion (DTI) | SWAP | Pattern generation |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 | 8 | 24-bit | Yes | Yes | Yes | Yes | Yes | Yes |
| 1 | 4 | 8 | 24-bit | Yes | Yes | Yes | Yes | Yes | Yes |
| 2 | 3 | 3 | 16-bit | Yes | - | Yes | - | - | - |
| 3 | 2 | 2 | 16-bit | Yes | - | - | - | - | - |
| 4 | 2 | 2 | 16-bit | Yes | - | - | - | - | - |

**Note:** The number of CC registers (CC_NUM) for each TCC corresponds to the number of compare/capture channels, so that a TCC can have more Waveform Outputs (WO_NUM) than CC registers.

### 6.2.8. IOSET Configurations

The SAM D5x/E5x has multiple peripheral instances, mapped to different IO locations. Each peripheral IO location is called IOSET and for a given peripheral, signals from different IOSET cannot be mixed.

For a given peripheral with two pads PAD0 and PAD1:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Valid: PAD0 and PAD1 in the same IOSETn.
- Invalid: PAD0 in IOSETx and PAD1 in IOSETy.

#### 6.2.8.1. SERCOM IOSET Configurations

The following tables lists each IOSET Pins for each SERCOM instance.

**Table 6-10.** SERCOM0 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs |
|---|---|---|---|---|
| PAD0 | PA08 | PB24 | PA04 | PC17 |
| PAD1 | PA09 | PB25 | PA05 | PC16 |
| PAD2 | PA10 | PC24 | PA06 | PC18 |
| PAD3 | PA11 | PC25 | PA07 | PC19 |

**Table 6-11.** SERCOM1 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs |
|---|---|---|---|---|
| PAD0 | PA16 | PC22 | PC27 | PA00 |
| PAD1 | PA17 | PC23 | PC28 | PA01 |
| PAD2 | PA18 | PD20 | PB22 | PA30 |
| PAD3 | PA19 | PD21 | PB23 | PA31 |

**Table 6-12.** SERCOM2 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs |
|---|---|---|---|---|
| PAD0 | PA12 | PB26 | PA09 | PB25 |
| PAD1 | PA13 | PB27 | PA08 | PB24 |
| PAD2 | PA14 | PB28 | PA10 | PC24 |
| PAD3 | PA15 | PB29 | PA11 | PC25 |

**Table 6-13.** SERCOM3 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs |
|---|---|---|---|---|
| PAD0 | PA22 | PB20 | PA17 | PC23 |
| PAD1 | PA23 | PB21 | PA16 | PC22 |
| PAD2 | PA24 | PA20 | PA18 | PD20 |
| PAD3 | PA25 | PA21 | PA19 | PD21 |

**Table 6-14.** SERCOM4 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs |
|---|---|---|---|---|
| PAD0 | PB12 | PB08 | PA13 | PB27 |
| PAD1 | PB13 | PB09 | PA12 | PB26 |
| PAD2 | PB14 | PB10 | PA14 | PB28 |
| PAD3 | PB15 | PB11 | PA15 | PB29 |

**Table 6-15.** SERCOM5 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs | IOSET 5 PINs | IOSET 6 PINs |
|---|---|---|---|---|---|---|
| PAD0 | PB16 | PA23 | PA23 | PA23 | PB31 | PB02 |
| PAD1 | PB17 | PA22 | PA22 | PA22 | PB30 | PB03 |
| PAD2 | PB18 | PA20 | PA24 | PB22 | PB00 | PB00 |
| PAD3 | PB19 | PA21 | PA25 | PB23 | PB01 | PB01 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-16.** SERCOM6 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs | IOSET 5 PINs |
|---|---|---|---|---|---|
| PAD0 | PC16 | PC04 | PD09 | PC13 | PC13 |
| PAD1 | PC17 | PC05 | PD08 | PC12 | PC12 |
| PAD2 | PC18 | PC06 | PD10 | PC14 | PC10 |
| PAD3 | PC19 | PC07 | PD11 | PC15 | PC11 |

**Table 6-17.** SERCOM7 IO SET Configuration

| SERCOM Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs | IOSET 5 PINs |
|---|---|---|---|---|---|
| PAD0 | PC12 | PD08 | PC12 | PB21 | PB30 |
| PAD1 | PC13 | PD09 | PC13 | PB20 | PB31 |
| PAD2 | PC14 | PD10 | PC10 | PB18 | PA30 |
| PAD3 | PC15 | PD11 | PC11 | PB19 | PA31 |

#### 6.2.8.2. GMAC IOSET Configurations

The following tables lists each IOSET pins for GMDIO and GMDC signals. All other GMAC signals can be used with all available IOSET configurations.

**Table 6-18.** GMAC IO SET Configuration

| GMAC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs |
|---|---|---|---|---|
| GMDC | PB14 | PC11 | PC22 | PA20 |
| GMDIO | PB15 | PC12 | PC23 | PA21 |

#### 6.2.8.3. I²S Configurations

The following tables lists each IOSET Pins for I²S instance.

**Table 6-19.** I²S IO SET Configuration

| I²S Signal | IOSET 1 PINs | IOSET 2 PINs |
|---|---|---|
| MCK0 | PA08 | PB17 |
| FS0 | PA09 | PA20 |
| SCK0 | PA10 | PB16 |
| SDO | PA11 | PA21 |
| SDI | PB10 | PA22 |
| FS1 | PB11 | PA23 |
| SCK1 | PB12 | PB28 |
| MCK1 | PB13 | PB29 |

#### 6.2.8.4. TC IOSET Configurations

The following tables lists each IOSET Pins for each TC instance.

**Table 6-20.** TC0 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs |
|---|---|---|---|
| WO0 | PA04 | PA08 | PB30 |
| WO1 | PA05 | PA09 | PB31 |

**Table 6-21.** TC1 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs |
|---|---|---|
| WO0 | PA06 | PA10 |
| WO1 | PA07 | PA11 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-22.** TC2 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs |
|---|---|---|---|
| WO0 | PA00 | PA12 | PA16 |
| WO1 | PA01 | PA13 | PA17 |

**Table 6-23.** TC3 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs |
|---|---|---|
| WO0 | PA14 | PA18 |
| WO1 | PA15 | PA19 |

**Table 6-24.** TC4 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs |
|---|---|---|---|
| WO0 | PB08 | PB12 | PA22 |
| WO1 | PB09 | PB13 | PA23 |

**Table 6-25.** TC5 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs |
|---|---|---|---|
| WO0 | PB10 | PB14 | PA24 |
| WO1 | PB11 | PB15 | PA25 |

**Table 6-26.** TC6 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs |
|---|---|---|---|
| WO0 | PB16 | PA30 | PB02 |
| WO1 | PB03 | PB17 | PA31 |

**Table 6-27.** TC7 IOSET Configuration

| TC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs |
|---|---|---|---|
| WO0 | PA20 | PB22 | PB00 |
| WO1 | PA21 | PB23 | PB01 |

### 6.2.8.5. TCC IOSET Configurations

The following tables lists each IOSET Pins for each TCC instance.

**Table 6-28.** TCC0 IO SET Configuration

| TCC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs | IOSET 5 PINs | IOSET 6 PINs |
|---|---|---|---|---|---|---|
| WO0 | PA08 | PC04 | PC10 | PC16 | PB12 | PA20 |
| WO1 | PA09 | PD08 | PC11 | PC17 | PB13 | PA21 |
| WO2 | PA10 | PD09 | PC12 | PC18 | PB14 | PA22 |
| WO3 | PA11 | PD10 | PC13 | PC19 | PB15 | PA23 |
| WO4 | PB10 | PD11 | PC14 | PC20 | PA16 | PB16 |
| WO5 | PB11 | PD12 | PC15 | PC21 | PA17 | PB17 |
| WO6 | PA12 | PC22 | PA18 | PB30 | N/A(1) | N/A(1) |
| WO7 | PA13 | PC23 | PA19 | PB31 | N/A(1) | N/A(1) |

**Note:** 1. The signal is available, but the edges are not aligned wrt. the other signals as specified.

**Table 6-29.** TCC1 IO SET Configuration

| TCC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs | IOSET 5 PINs |
|---|---|---|---|---|---|
| WO0 | PA16 | PD20 | PB18 | PB10 | PC14 |
| WO1 | PA17 | PD21 | PB19 | PB11 | PC15 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-29.** TCC1 IO SET Configuration (continued)

| TCC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs | IOSET 5 PINs |
|---|---|---|---|---|---|
| WO2 | PA18 | PB20 | PB26 | PA12 | PA14 |
| WO3 | PA19 | PB21 | PB27 | PA13 | PA15 |
| WO4 | PA20 | PB28 | PA08 | PC10 | N/A(1) |
| WO5 | PA21 | PB29 | PA09 | PC11 | N/A(1) |
| WO6 | PA22 | PA10 | PC12 | N/A(1) | N/A(1) |
| WO7 | PA23 | PA11 | PC13 | N/A(1) | N/A(1) |

**Note:** 1. The signal is available, but the edges are not aligned wrt. the other signals as specified.

**Table 6-30.** TCC2 IO SET Configuration

| TCC Signal | IOSET 1 PINs | IOSET 2 PINs |
|---|---|---|
| WO0 | PA14 | PA30 |
| WO1 | PA15 | PA31 |
| WO2 | PA24 | PB02 |

**Table 6-31.** TCC3 IO SET Configuration

| TCC Signal | IOSET 1 PINs | IOSET 2 PINs |
|---|---|---|
| WO0 | PB12 | PB16 |
| WO1 | PB13 | PB17 |

**Table 6-32.** TCC4 IO SET Configuration

| TCC Signal | IOSET 1 PINs | IOSET 2 PINs |
|---|---|---|
| WO0 | PB14 | PB30 |
| WO1 | PB15 | PB31 |

### 6.2.8.6. PDEC IOSET Configurations

The following tables lists each IOSET Pins for PDEC instance.

**Table 6-33.** PDEC IO SET Configuration

| PDEC Signal | IOSET 1 PINs | IOSET 2 PINs | IOSET 3 PINs | IOSET 4 PINs |
|---|---|---|---|---|
| PDEC[0] | PC16 | PB18 | PA24 | PB23 |
| PDEC[1] | PC17 | PB19 | PA25 | PB24 |
| PDEC[2] | PC18 | PB20 | PB22 | PB25 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 6.2.9. GPIO Clusters

**Table 6-34.** GPIO Clusters [1,2]

| Package | Cluster | GPIO |
|---|---|---|
| 128pins | VDDIOB | PA11, PA10, PA09, PA08 |
| | | PB11, PB10 |
| | | PC07, PC06, PC05, PC04 |
| | VDDIO | PA31, PA30, PA27, PA25, PA24, PA23, PA22, PA21, PA20, PA19, PA18, PA17, PA16, PA15, PA14, PA13, PA12 |
| | | PB31, PB30, PB29, PB28, PB27, PB26, PB25, PB24, PB23, PB22, PB21, PB20, PB19, PB18, PB17, PB16, PB15, PB14, PB13, PB12, |
| | | PC31, PC30, PC28, PC27, PC26, PC25, PC24, PC23, PC22, PC21, PC20, PC19, PC18, PC17, PC16, PC15, PC14, PC13, PC12, PC11, PC10 |
| | | PD21, PD20, PD12, PD11, PD10, PD09, PD08 |
| | VDDANA | PA07, PA06, PA05, PA04, PA03, PA02 |
| | | PB09, PB08, PB07, PB06, PB05, PB04 |
| | | PC03, PC02 |
| | | PD01, PD00 |
| | VSWOUT | PA01, PA00 |
| | | PB03, PB02, PB01, PB00 |
| | | PC01, PC00 |
| 120pins | VDDIOB | PA11, PA10, PA09, PA08 |
| | | PB11, PB10 |
| | | PC07, PC06, PC05, PC04 |
| | VDDIO | PA31, PA30, PA27, PA25, PA24, PA23, PA22, PA21, PA20, PA19, PA18, PA17, PA16, PA15, PA14, PA13, PA12 |
| | | PB31, PB30, PB29, PB28, PB27, PB26, PB25, PB24, PB23, PB22, PB21, PB20, PB19, PB18, PB17, PB16, PB15, PB14, PB13, PB12, |
| | | PC31, PC30, PC28, PC27, PC26, PC25, PC24, PC23, PC22, PC21, PC20, PC19, PC18, PC17, PC16, PC15, PC14, PC13, PC12, PC11, PC10 |
| | | PD21, PD20, PD12, PD11, PD10, PD09, PD08 |
| | VDDANA | PA07, PA06, PA05, PA04, PA03, PA02 |
| | | PB09, PB08, PB07, PB06, PB05, PB04 |
| | | PC03, PC02 |
| | | PD01, PD00 |
| | VSWOUT | PA01, PA00 |
| | | PB03, PB02, PB01, PB00 |
| | | PC01, PC00 |
| 100pins | VDDIOB | PA11, PA10, PA09, PA08 |
| | | PB11, PB10 |
| | | PC07, PC06, PC05 |
| | VDDIO | PA31, PA30, PA27, PA25, PA24, PA23, PA22, PA21, PA20, PA19, PA18, PA17, PA16, PA15, PA14, PA13, PA12 |
| | | PB31, PB30, PB25, PB24, PB23, PB22, PB21, PB20, PB19, PB18, PB17, PB16, PB15, PB14, PB13, PB12 |
| | | PC28, PC27, PC26, PC25, PC24, PC21, PC20, PC19, PC18, PC17, PC16, PC15, PC14, PC13, PC12, PC11, PC10 |
| | VDDANA | PA07, PA06, PA05, PA04, PA03, PA02 |
| | | PB09, PB08, PB07, PB06, PB05, PB04 |
| | | PC03, PC02 |
| | VSWOUT | PA01, PA00 |
| | | PB03, PB02, PB01, PB00 |
| | | PC01, PC00 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 6-34.** GPIO Clusters [1,2] (continued)

| Package | Cluster | GPIO |
|---|---|---|
| 64 Pins | VDDIOB | PA11, PA10, PA09, PA08 |
| | | PB11, PB10 |
| | VDDIO | PB12,PB13,PB14,PB15,PB16,PB17,PB30,PB31 |
| | | PA12,PA13,PA16,PA17,PA18,PA19, PA20, PA21,PA22,PA23,PA24,PA25,PA27,PA30,PA31 |
| | | PA14,PA15,PB22,PB23 |
| | VDDANA | PA2,PA3,PB4,PB5,PB6,PB7,PB8,PB9,PA4,PA5,PA6,PA7 |
| | VSWOUT | PB0,PB1,PB2,PB3,PA0,PA1 |
| 48 pins | VDDIO | PA8, PA9,PA10,PA11 |
| | | PB10,PB11,PA12,PA13,PA14,PA15 |
| | | PA16,PA17,PA18,P19,PA20,PA21,PA22,PA23,PA24,PA25 |
| | | PB22,PB23 |
| | | PA27 |
| | | PA30, PA31 |
| | VDDANA | PA2,PA3,PB8,PB9,PA4,PA5,PA6,PA7 |
| | VSWOUT | PB2,PB3,PA0,PA1 |

**Notes:**

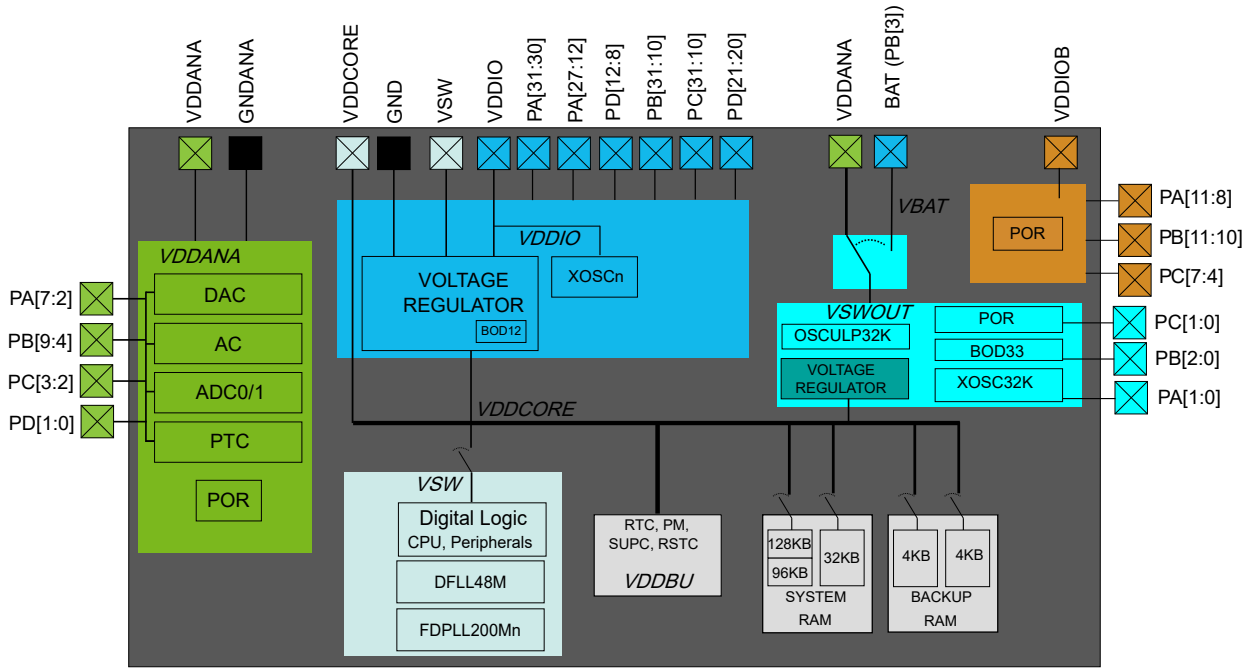
1. The RESETN pin in all packages are connected to the VDDIO cluster.
2. For VDDIO,VDDIOB and VDDANA refer to Table 54-1. Absolute Maximum Ratings.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 7. Power Supply and Start-Up Considerations

## 7.1. Power Domain Overview

**Figure 7-1.** Power Domain Block Diagram




The SAM D5x/E5x power domains are not independent of each other:

- *VDDCORE*, *VDDIO* and *VDDIOB* share GND, whereas *VDDANA* uses GNDANA.
- *VDDANA* and *VDDIO* must share the main supply, VDD.
- VDDCORE pin is just an output for monitoring the internal voltage regulator. This is not an input for an external supply.
- *VSWOUT*, *VSW* and *VDDBU* are internal power domains.
- The VSW pin is for inductor connection to run the Main Voltage Regulator in switching mode.

## 7.2. Power Supply Considerations

### 7.2.1. Power Supplies

The SAM D5x/E5x has the following power supply pins:

- VDDIO – Powers I/O lines, XOSCn and the internal regulator for VDDCORE. Voltage is 1.71V to 3.63V.
- VDDIOB – Powers I/O B lines. Voltage is 1.71V to 3.63V.
- VDDANA – Powers I/O lines, the Automatic Power Switch, ADC0/1, AC, DAC and PTC. Voltage is 1.71V to 3.63V.
- VBAT – Powers the Automatic Power Switch. Voltage is 1.71V to 3.63V
- VDDCORE – Serves as the internal voltage regulator output in linear mode, depending on the powering configuration. It powers the VSW core power domain and the VDDBU backup domain, memories, peripherals, DFLL48M, FDPLL200Mn, and RAMs. Voltage is 1.2V typical.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The Automatic Power Switch is a configurable switch that selects between VDD and VBAT as supply for the internal output VSWOUT, see the figure in Power Domain Overview.

The same voltage must be applied to both VDDIO and VDDANA. In this data sheet, this common voltage applied to both VDDIO and VDDANA, is represented as VDD.

VDDIOB voltage level must be equal or lower than VDDIO.

The ground pins, GND, are common to VDDCORE, and VDDIO. The ground pin for VDDANA is GNDANA.

For decoupling recommendations for the different power supplies, refer to the schematic checklist.

**Related Links**

Schematic Checklist
GPIO Clusters
Typical Powering Schematic

### 7.2.2. Voltage Regulator

The SAM D5x/E5x internal Main Voltage Regulator has three different modes:

- Linear mode: This is the default mode when CPU and peripherals are running. It does not require an external inductor.
- Switching mode. This is the most power efficient mode when the CPU and peripherals are running and which requires an external inductor between VSW and VDDCORE. This mode can be selected by software on the fly.
- Shutdown mode. When the chip is in backup mode, the internal regulator is off, the VSW core power domain is OFF. The VDDBU backup domain is powered by the backup regulator (LPVREG).

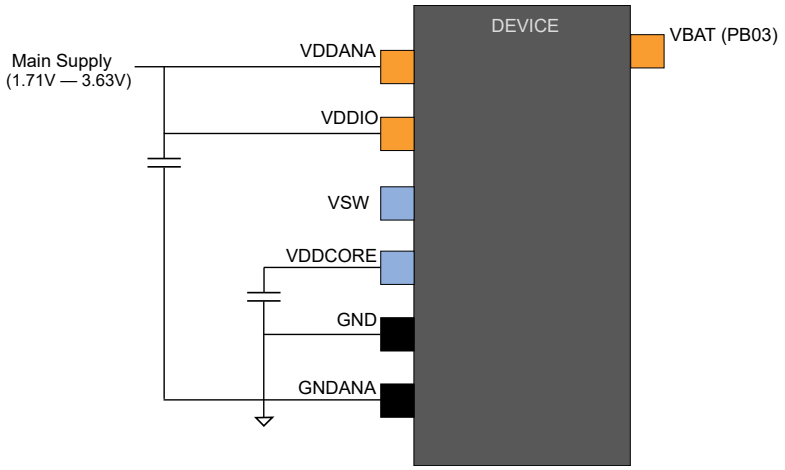
Note that the Voltage Regulator modes are controlled by the Power Manager.

### 7.2.3. Typical Powering Schematic

The SAM D5x/E5x uses a single supply from 1.71V to 3.63V.

The following figure shows the recommended power supply connection.

**Figure 7-2.** Power Supply Connection for Linear Mode Only





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 7-3.** Power Supply Connection for Switching/Linear Mode

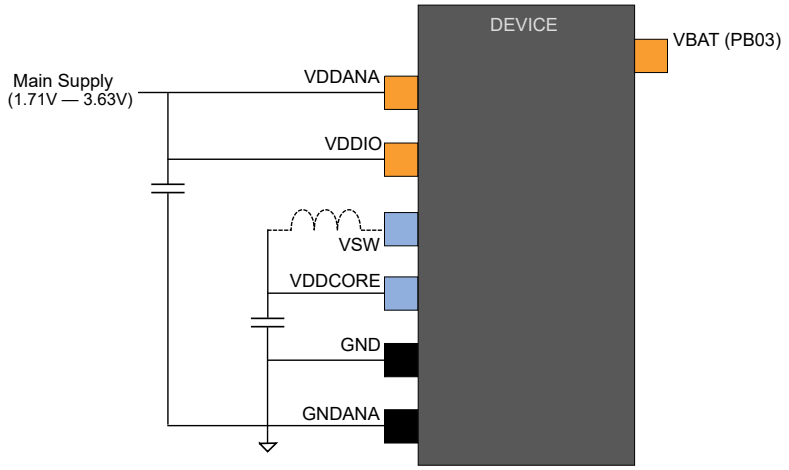
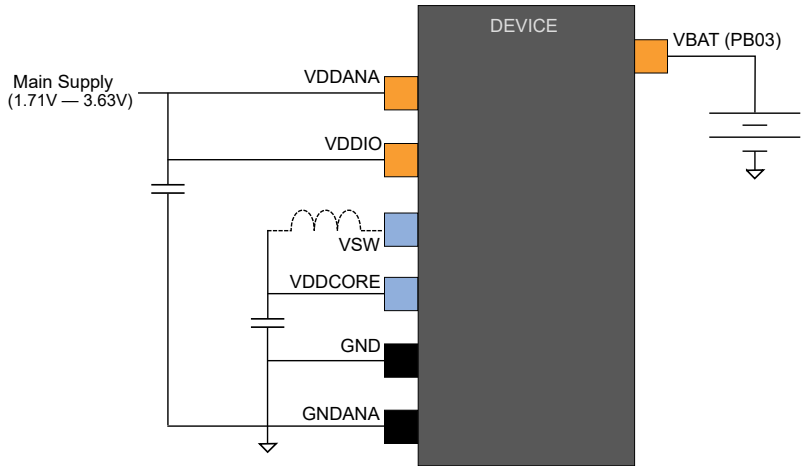



**Figure 7-4.** Power Supply Connection for Battery Backup




### 7.2.4. Power-Up Sequence

#### 7.2.4.1. Supply Order

VDDIO and VDDANA must have the same supply sequence, and must be connected together.

Note that VDDIO supplies the XOSCn, so VDDIO must be present before the applicaion uses the XOSC feature. This is also applicable to all digital features present on pins supplied by VDDIO. VDDIOB must be present before the application uses features present on pins supplied by VDDIOB.

#### 7.2.4.2. Minimum Rise Rate

One integrated power-on reset (POR) circuits monitoring VDDANA requires a minimum rise rate. See Supply Characteristics.

#### 7.2.4.3. Maximum Rise Rate

The rise rate of the power supplies must not exceed the values described in Electrical Characteristics. See Supply Characteristics.

## 7.3. Power-Up

This section summarizes the power-up sequence of the SAM D5x/E5x. The behavior after power-up is controlled by the Power Manager.

**Related Links**

PM – Power Manager


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 7.3.1. Starting of Internal Regulator

After power-up, the device is set to its initial state and kept in Reset, until the power has stabilized throughout the device.

The internal regulator provides VDDCORE. Once the external voltage VDDIO/VDDANA and the internal VDDCORE reach a stable value, the internal Reset is released.

**Related Links**

PM – Power Manager

### 7.3.2. Starting of Clocks

Once the power has stabilized and the internal Reset is released, the device will use a 48MHz clock by default. The clock source for this clock signal is DFLL48M running in open loop mode, which is enabled after a reset by default. This is also the default time base for Generic Clock Generator 0. In turn, Generator 0 provides the main clock GCLK_MAIN which is used by the Main Clock module (MCLK).

Some synchronous system clocks are active after Start-Up, allowing software execution. Refer to the “Clock Mask Registers” section in the MCLK-Main Clock documentation for the list of clocks that are running by default. Synchronous system clocks that are running receive the 48MHz clock from Generic Clock Generator 0. Other generic clocks are disabled.

**Related Links**

PM – Power Manager

### 7.3.3. I/O Pins

After power-up, the I/O pins are tri-stated except: PA24, PA25 and PA30. Pins PA24, PA25 are configured in input pull-down mode. PA30 is configured in input pull up mode in order to serve as part of the debug interface.

### 7.3.4. Fetching of Initial Instructions

After Reset has been released, the CPU starts fetching PC and SP values from the Reset address, 0x00000000. This points to the first executable address in the internal Flash memory. The code read from the internal Flash can be used to configure the clock system and clock sources. See the related peripheral documentation for details. Refer to the ARM Architecture Reference Manual for more information on CPU startup (www.arm.com).

## 7.4. Power-On Reset and Brown-Out Detector

The SAM D5x/E5x embeds three features to monitor, warn and/or reset the device:

- POR: Power-on Reset on the main supply VDD (VDDANA/VDDIO) and VSWOUT
- BOD33: Brown-out detector on VDD or VBAT
- Brown-out detector internal to the voltage regulator for VDDCORE. BOD12 is calibrated in production and its calibration parameters are stored in the NVM User Row. This data should not be changed if the User Row is written to in order to assure correct behavior.

### 7.4.1. Power-On Reset on VSWOUT

VSWOUT is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VSWOUT goes below the threshold voltage, the entire chip is reset.

### 7.4.2. Power-On Reset on the main supply VDD (VDDANA/VDDIO)

The Main supply VDD (VDDANA/VDDIO) is monitored by POR. Monitoring is always activated, including startup and all sleep modes. If VDD goes below the threshold voltage, all I/Os supplied by VDDIO are reset.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 7.4.3. Brown-Out Detector on VSWOUT/VBAT

BOD33 monitors VSWOUT or VBAT depending on configuration.

**Related Links**

SUPC – Supply Controller

### 7.4.4. Brown-Out Detector on VDDCORE

Once the device has started up, BOD12 monitors the internal VDDCORE.

**Related Links**

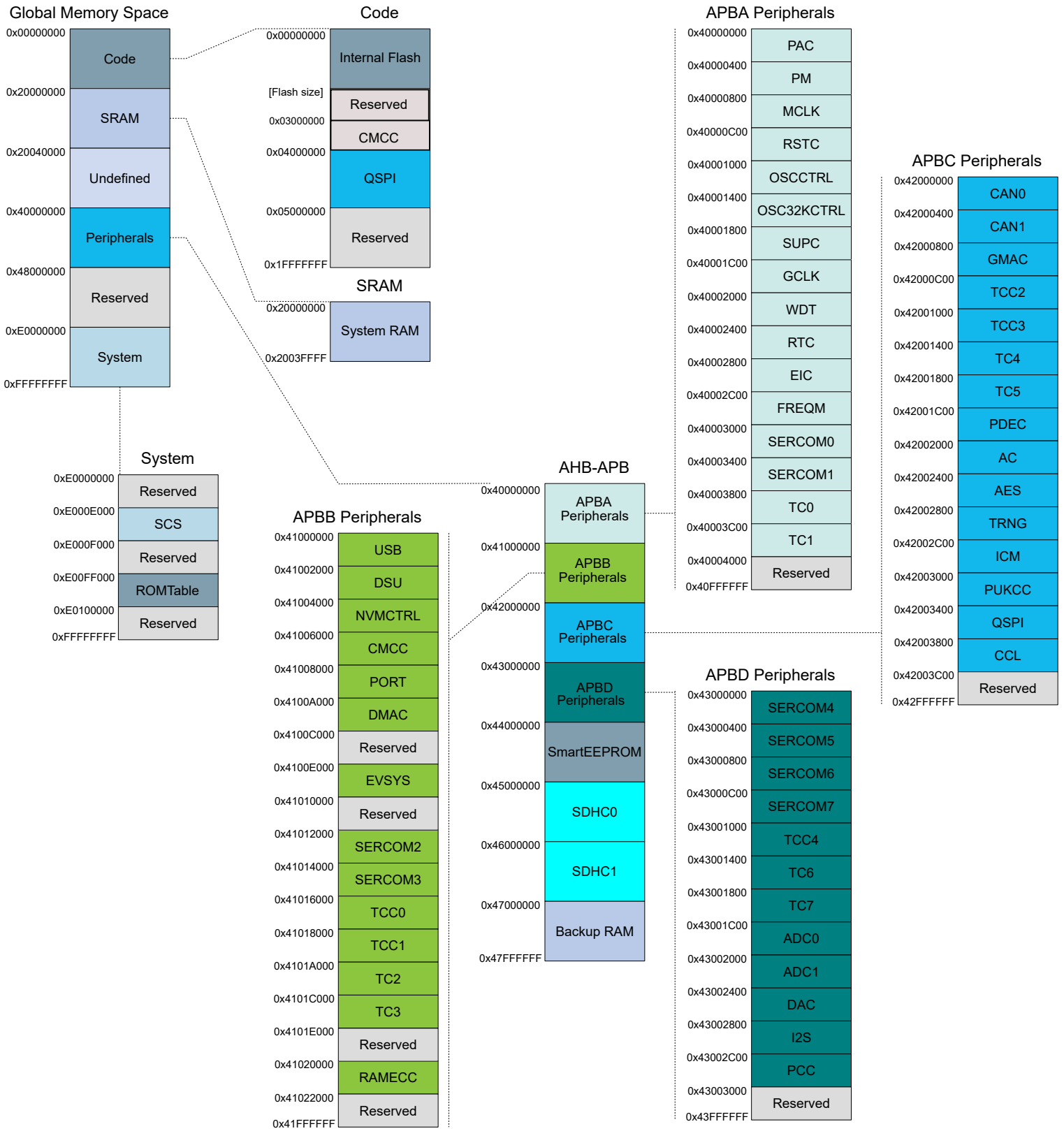
SUPC – Supply Controller


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

# 8. Product Memory Mapping Overview

**Figure 8-1.** Product Mapping





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Memories


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 9. Memories

## 9.1. Embedded Memories

- Internal high-speed Flash with Read-While-Write (RWW) capability on a section of the array
- Internal high-speed RAM, single-cycle access at full speed
- Internal backup RAM, single-cycle access at full speed

## 9.2. Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed, and they are never remapped in any way, even during boot. The 32-bit physical address space is mapped as follows:

**Table 9-1.** Physical Memory Map

| Memory | Start Address | Size in KB (unless otherwise stated) | | |
|---|---|---|---|---|
| | | SAMD51x20 SAME51x20 SAME53x20 SAME54x20 | SAMD51x19 SAME51x19 SAME53x19 SAME54x19 | SAMD51x18 SAME51x18 SAME53x18 |
| Embedded Flash | 0x00000000 | 1024 | 512 | 256 |
| Embedded SRAM | 0x20000000 | 256 | 192 | 128 |
| Peripheral Bridge A | 0x40000000 | 16384 Bytes | | |
| Peripheral Bridge B | 0x41000000 | | | |
| Peripheral Bridge C | 0x42000000 | | | |
| Peripheral Bridge D | 0x43000000 | | | |
| Backup SRAM | 0x47000000 | 8 | | |
| NVM User Row | 0x00804000 | 512 Bytes | | |

**Note:**

1. X = G, J, N or P. Refer to Ordering Information for available device part numbers.

### 9.2.1. Flash Memory Parameters

A single page contains 512 Bytes, which is applicable to all the device part numbers listed in the Configuration Summary.

Number of pages available in a device part number will vary depending on available maximum Flash memory size.

**Equation 9-1.** Calculating Flash Memory

$$NumberofPages = \frac{FlashSize(Bytes)}{PageSize(Bytes)}$$

## 9.3. SRAM Memory Configuration

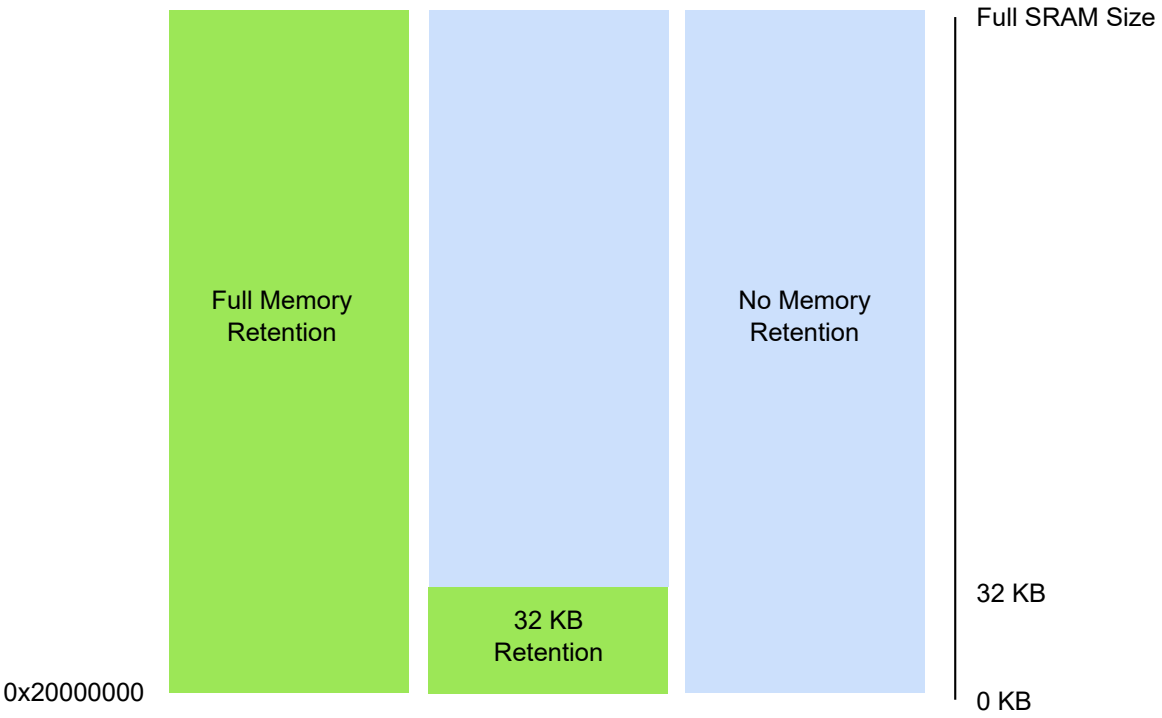
**Retention**

Depending on the application and power budget needs, part of the system memory can be retained in Standby or Hibernate sleep modes. The amount of the SRAM retained in this mode is software selectable, by writing the RAMCFG bits in the Power Manager Standby Configuration register and Hibernate Configuration register respectively (STDBYCFG.RAMCFG and HIBCFG.RAMCFG).

By default, the entire system memory section is retained, but no retention or bottom 32KB memory retention options are also available.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 9-1.** Retention Options



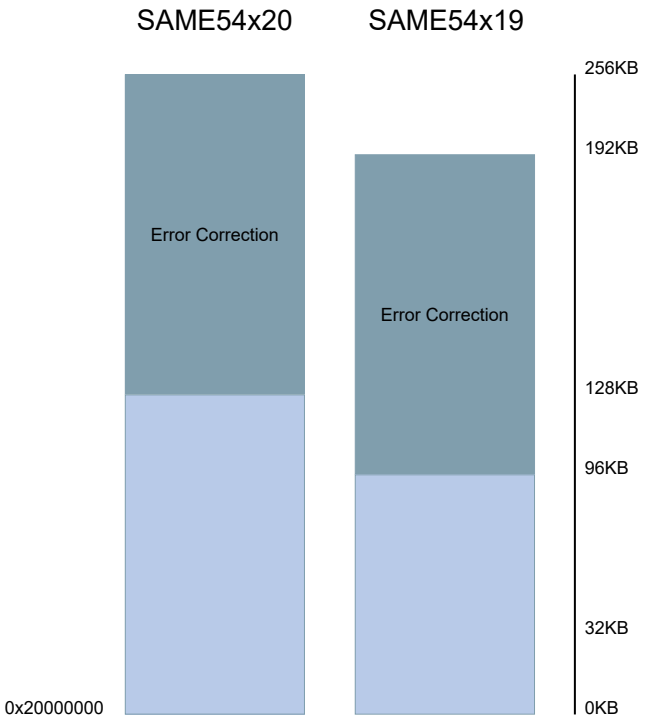

### RAM Error Correction

For safety applications, the SAM D5x/E5x family embeds error correction codes (ECC) to detect and correct single bit errors, or to enable dual error detection for the system memory. The ECC is software selectable through the RAM ECCDIS bit in the NVM User Row. For additional information, refer to Table 9-2.

When enabled, the top half system memory will be reserved to store the ECC, and will not be available for the application.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 9-2.** Memory with RAM Error Correction



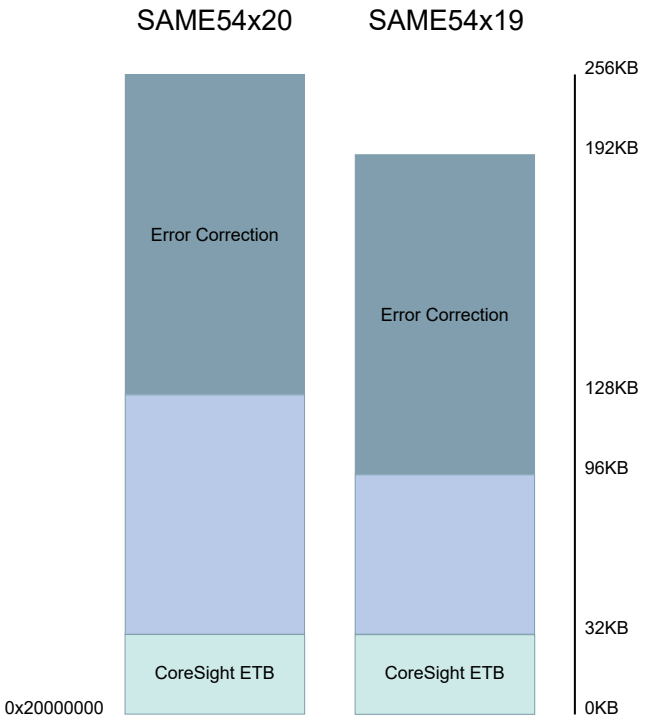

**Note:** If the ECC is used, full SRAM retention must be enabled.

### CoreSight ETB Connection

When enabled, the bottom 32 KB system memory space is reserved for CoreSight ETB debug usage. The figure below shows an example where both ECC and CoreSight ETB are enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 9-3.** Memory with ECC and CoreSight ETB




## 9.4. NVM User Row Mapping (UROW)

The NVM User Row can be read at address 0x00804000. The size of the NVM User Row is 512 Bytes.

The first eight 32-bit words (32 Bytes) of the Non Volatile Memory (NVM) User Row contain calibration data that are automatically read at device power on. The remaining 480 Bytes can be used for storing custom parameters.

To write the NVM User Row, refer to the NVMCTRL (Non-Volatile Memory Controller) documentation.

When writing to the user pages, the new values do not get loaded by the other peripheral on the device until a device reset occurs.

**Note:** Before erasing the NVM User Row, ensure that the first 32 Bytes are read to a buffer and later written back to the same area unless a configuration change is intended.

**Table 9-2.** NVM User Row Mapping - Dedicated Entries

| Bit Pos. | Name | Usage | Related Peripheral Register | Default Values |
|---|---|---|---|---|
| 0 | BOD33 Disable | BOD33 Disable at power-on. | SUPC.BOD33 | 0x1 |
| 8:1 | BOD33 Level | BOD33 threshold level at power-on. | SUPC.BOD33 | 0x1C |
| 10:9 | BOD33 Action | BOD33 Action at power-on. | SUPC.BOD33 | 0x1 |
| 14:11 | BOD33 Hysteresis | BOD33 Hysteresis configuration at power-on. | SUPC.BOD33 | 0x2 |
| 25:15 | BOD12 Calibration Parameters | Factory settings - do not change.(1) | | - |
| 29:26 | NVM BOOTPROT | NVM Bootloader Size: (15-BOOTPROT)*8KB | NVMCTRL.STATUS | 0xF |
| 31:30 | Reserved | Factory settings - do not change. | | - |
| 35:32 | SBLK | Number of NVM Blocks composing a SmartEEPROM sector | NVMCTRL.SEESTAT | 0x0 |
| 38:36 | PSZ | SmartEEPROM Page Size | NVMCTRL.SEESTAT | 0x0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 9-2.** NVM User Row Mapping - Dedicated Entries (continued)

| Bit Pos. | Name | Usage | Related Peripheral Register | Default Values |
|---|---|---|---|---|
| 39 | RAM ECCDIS | RAM ECC Disable | RAMECC | 0x1 |
| 47:40 | Reserved | Factory settings - do not change. | | - |
| 48 | WDT Enable | WDT Enable at power-on. | WDT.CTRLA | 0x0 |
| 49 | WDT Always-On | WDT Always-On at power-on. | WDT.CTRLA | 0x0 |
| 53:50 | WDT Period | WDT Period at power-on. | WDT.CONFIG | 0xB |
| 57:54 | WDT Window | WDT Window mode time-out at power-on. | WDT.CONFIG | 0xB |
| 61:58 | WDT EWOFFSET | WDT Early Warning Interrupt Time Offset at power-on. | WDT.EWCTRL | 0xB |
| 62 | WDT WEN | WDT Window Mode Enable at power-on. | WDT.CTRLA | 0x0 |
| 63 | Reserved | Factory settings - do not change. | | |
| 95:64 | NVM LOCKS | NVM Region Lock Bits. | NVMCTRL.RUNLOCK | 0xFFFF FFFF |
| 127:96 | (fourth word) | User page | | |
| 159:128 | Reserved | Factory settings - do not change. | | |
| Other | - | User pages | | |

**CAUTION**

1. BOD12 is calibrated in production, and the calibration parameters must not be changed to ensure the correct device behavior.

**Related Links**

NVMCTRL – Nonvolatile Memory Controller
SUPC – Supply Controller
BOD33
3.3V Brown-Out Detector (BOD33) Control
WDT – Watchdog Timer
CTRLA
Control A
CONFIG
Configuration
EWCTRL
Early Warning Control
Device Temperature Measurement

## 9.5. NVM Software Calibration Area Mapping (SW0)

The NVM Software Calibration Area contains calibration data that are determined and written during production test. These calibration values should be read by the application software and written back to the corresponding register.

The NVM Software Calibration Area can be read at address 0x00800080.

The NVM Software Calibration Area can not be written.

**Table 9-3.** NVM Software Calibration Area Mapping

| Bit Position | Name | Description | Default Value |
|---|---|---|---|
| 1:0 | AC BIAS | AC Comparator 0/1 Bias Scaling. To be written to the AC CALIB register. | 0x1 |
| 4:2 | ADC0 BIASCOMP | Bias comparator scaling. To be written to the ADC0 CALIB register. | 0x7 |
| 7:5 | ADC0 BIASREFBUF | Bias reference buffer scaling. To be written to the ADC0 CALIB register. | 0x7 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 9-3.** NVM Software Calibration Area Mapping (continued)

| Bit Position | Name | Description | Default Value |
|---|---|---|---|
| 10:8 | ADC0 BIASR2R | Bias rail-to-rail amplifier scaling. To be written to the ADC0 CALIB register. | 0x7 |
| 15:11 | Reserved | - | - |
| 18:16 | ADC1 BIASCOMP | Bias comparator scaling. To be written to the ADC1 CALIB register. | 0x7 |
| 21:19 | ADC1 BIASREFBUF | Bias reference buffer scaling. To be written to the ADC1 CALIB register. | 0x7 |
| 24:22 | ADC1 BIASR2R | Bias rail-to-rail amplifier scaling. To be written to the ADC1 CALIB register. | 0x7 |
| 35:25 | Reserved | - | - |
| 36:32 | USB TRANSN | USB TRANSN calibration value. To be written to the USB PADCAL register. | 0x09 |
| 41:37 | USB TRANSP | USB TRANSP calibration value. To be written to the USB PADCAL register. | 0x19 |
| 44:42 | USB TRIM | USB TRIM calibration value. To be written to the USB PADCAL register. | 0x6 |

The NVM Software Calibration Area for temperature calibration parameters can not be written.

The NVM Software Calibration Area for temperature calibration parameters can be read at address 0x00800100.

**Table 9-4.** NVM Software Calibration Area Mapping - Temperature Calibration Parameters

| Bit Position | Name | Description |
|---|---|---|
| 7:0 | TLI | Integer part of calibration temperature TL |
| 11:8 | TLD | Decimal part of calibration temperature TL |
| 19:12 | THI | Integer part of calibration temperature TH |
| 23:20 | THD | Decimal part of calibration temperature TH |
| 39:24 | Reserved | Reserved for future use. |
| 51:40 | VPL | Temperature calibration parameters. |
| 63:52 | VPH | |
| 75:64 | VCL | |
| 87:76 | VCH | |
| 127:88 | Reserved | Reserved for future use. |

**Note:** Engineering Sample devices have no valid temperature calibration parameters.

## 9.6. Serial Number

Each device has a unique 128-bit serial number which is a concatenation of four 32-bit words contained at the following addresses:

Word 0: 0x008061FC

Word 1: 0x00806010

Word 2: 0x00806014

Word 3: 0x00806018

The uniqueness of the serial number is guaranteed only when using all 128 bits.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 10. Processor and Architecture

## 10.1. Cortex M4 Processor

The ARM®Cortex™-M4 processor is a high performance 32-bit processor designed for the microcontroller market. It offers the following significant benefits to developers:

- Outstanding processing performance combined with fast interrupt handling
- Enhanced system debug with extensive breakpoint and trace capabilities
- Efficient processor core, system, and memories
- Ultra low-power consumption with integrated sleep modes
- Platform security robustness, with integrated memory protection unit (MPU).

The implemented ARM Cortex-M4 is revision r0p1

For additional information, refer to http://www.arm.com

The Cortex-M4 processor is built on a high-performance processor core with a 3-stage pipeline Harvard architecture, making it ideal for demanding embedded applications. The processor delivers exceptional power efficiency through an efficient instruction set and extensively optimized design, providing high-end processing hardware including IEEE 754-compliant single-precision floating-point computation, a range of single-cycle and SIMD multiplication and multiply-with-accumulate capabilities, saturating arithmetic, and dedicated hardware division.

To facilitate the design of cost-sensitive devices, the Cortex-M4 processor implements tightly-coupled system components that reduce processor area while significantly improving interrupt handling and system debug capabilities. The Cortex-M4 processor implements a version of the Thumb instruction set based on Thumb®-2 technology, ensuring high code density and reduced program memory requirements. The Cortex-M4 instruction set provides the exceptional performance expected of a modern 32-bit architecture, with the high code density of 8-bit and 16-bit microcontrollers.

The Cortex-M4 processor closely integrates a configurable *Nested Vector Interrupt Controller* (NVIC), to deliver industry-leading interrupt performance. The NVIC includes a *Non-Maskable interrupt* (NMI), and provides up to 8 interrupt priority levels. The tight integration of the processor core and NVIC provides fast execution of *Interrupt Service Routines* (ISRs), dramatically reducing interrupt latency. This is achieved through the hardware stacking of registers, and the ability to suspend load-multiple and store-multiple operations. Interrupt handlers do not require wrapping in assembler code, removing any code overhead from the ISRs. A tail-chain optimization also significantly reduces the overhead when switching from one ISR to another.

To optimize low-power designs, the NVIC integrates with the sleep modes, that include a deep sleep function that enables the entire device to be rapidly powered down while still retaining program state.

### 10.1.1. System Level Interface

The Cortex-M4 processor provides multiple interfaces using AMBA technology to provide high-speed, low-latency memory accesses. It supports unaligned data accesses and implements atomic bit manipulation that enables faster peripheral controls, system spinlocks, and thread-safe Boolean data handling.

The Cortex-M4 processor has a Memory Protection Unit (MPU) that provides fine grain memory control, enabling applications to utilize multiple privilege levels, separating and protecting code, data, and stack on a task-by-task basis. Such requirements are becoming critical in many embedded applications such as automotive.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 10.1.2. Integrated Configurable Debug

The Cortex-M4 processor implements a complete hardware debug solution. This provides high system visibility of the processor and memory through a 2-pin *Serial Wire Debug* (SWD) port that is ideal for microcontrollers and other small package devices.

For system trace the processor integrates an *Instrumentation Trace Macrocell* (ITM) alongside data watchpoints and a profiling unit. The *Embedded Trace Macrocell* (ETM) delivers unrivaled instruction trace capture in an area far smaller than traditional trace units, enabling many low cost MCUs to implement full instruction trace for the first time.

To enable simple and cost-effective profiling of the system events these generate, a stream of software-generated messages, data trace, and profiling information is exported over three different ways:

- Output off chip using the TPIU, through a single pin, called *Serial Wire Viewer* (SWV). Limited to ITM system trace
- Output off chip using the TPIU, through a 4-bit pin interface. Bandwidth is limited
- Internally stored in RAM, using the CoreSight ETB. Bandwidth is then optimal but capacity is limited

The *Flash Patch and Breakpoint Unit* (FPB) provides up to 8 hardware breakpoint comparators that debuggers can use. The comparators in the FPB also provide remap functions of up to 8 words in the program code in the Code memory region. This enables applications stored on a non-erasable, ROM-based microcontroller to be patched if a small programmable memory, for example Flash, is available in the device. During initialization, the application in ROM detects, from the programmable memory, whether a patch is required. If a patch is required, the application programs the FPB to remap a number of addresses. When those addresses are accessed, the accesses are redirected to a remap table specified in the FPB configuration, which means the program in the non-modifiable ROM can be patched.

### 10.1.3. Cortex-M4 Processor Features and Configuration

- Thumb® instruction set combines high code density with 32-bit performance
- IEEE 754-compliant single-precision Floating Point Unit (FPU)
- Integrated sleep modes for low power consumption
- Fast code execution permits slower processor clock or increases Sleep mode time
- Hardware division and fast digital-signal-processing orientated multiply accumulate
- Saturating arithmetic for signal processing
- Deterministic, high-performance interrupt handling for time-critical applications
- Memory Protection Unit (MPU) for safety-critical applications
- Extensive debug and trace capabilities: Serial Wire Debug and Serial Wire Trace reduce the number of pins required for debugging, tracing, and code profiling.

| Features | Cortex-M4 Options | SAM D5x/E5x Configuration |
|---|---|---|
| Interrupts | 1 to 240 | 138 |
| Number of priority bits | 3 to 8 | 3 = eight levels of priority |
| Data endianness | Little-endian or big-endian | Little-endian |
| SysTick Timer calibration value | | 0x80000000 |
| MPU | Present or Not present | Present |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Cortex-M4 Processor Features and Configuration (continued) | | |
|---|---|---|
| **Features** | **Cortex-M4 Options** | **SAM D5x/E5x Configuration** |
| Debug support level | 0 = No debug. No DAP, breakpoints, watchpoints, Flash patch, or halting debug.<br>1 = Minimum debug. Two breakpoints, one watchpoint, no Flash patch.<br>2 = Full debug minus DWT data matching.<br>3 = Full debug plus DWT data matching. | 3 = Full debug plus DWT data matching. |
| Trace support level | 0 = No trace. No ETM, ITM or DWT triggers and counters.<br>1 = Standard trace. ITM and DWT triggers and counters, but no ETM.<br>2 = Full trace. Standard trace plus ETM.<br>3 = Full trace plus HTM port. | 2 = Full trace. ITM, TPIU, ETM, and DWT triggers and counters are present. HTM port is not present |
| JTAG | Present or Not present | Not present |
| Bit Banding | Present or Not present | Not present |
| FPU | Present or Not present | Present |

### 10.1.4. Cortex-M4 Core Peripherals

**Nested Vectored Interrupt Controller**
The Nested Vector Interrupt Controller (NVIC) is an embedded interrupt controller that supports low latency interrupt processing.

**System Control Block**
The System Control Block (SCB) is the programmer's model interface to the processor. It provides system implementation information and system control, including configuration, control, and reporting of system exceptions. Refer to the Cortex-M4 Technical Reference Manual for details (http://www.arm.com).

**System Timer**
The system timer, SysTick, is a 24-bit count-down timer. Use this as a Real-Time Operating System (RTOS) tick timer or as a simple counter. The SysTick timer runs on the processor clock and it does not decrement when the processor is halted for debugging. Refer to the Cortex-M4 Technical Reference Manual for details (http://www.arm.com).

**Memory Protection Unit**
The Memory Protection Unit (MPU) improves system reliability by defining the memory attributes for different memory regions. It provides up to eight different regions, and an optional predefined background region. Refer to the Cortex-M4 Technical Reference Manual for details (http://www.arm.com).

**Floating-Point Unit**
The Floating Point Unit (FPU) provides IEEE 754-compliant operations on single-precision, 32-bit, floating-point values. Refer to the Cortex-M4 Technical Reference Manual for details (http://www.arm.com).

### 10.1.5. Cortex-M4 Address Map

| Address | Core Peripheral |
|---|---|
| 0xE000E008-0xE000E00F | System control block |
| 0xE000E010-0xE000E01F | System timer |
| 0xE000E100-0xE000E4EF | Nested Vectored Interrupt Controller |
| 0xE000ED00-0xE000ED3F | System control block |
| 0xE000ED90-0xE000ED93 | MPU Type Register |
| 0xE000ED90-0xE000EDB8 | Memory Protection Unit |
| 0xE000EF00-0xE000EF03 | Nested Vectored Interrupt Controller |
| 0xE000EF30-0xE000EF44 | Floating Point Unit |

**Related Links**
Product Memory Mapping Overview


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 10.2. Nested Vector Interrupt Controller

### 10.2.1. Overview

The Nested Vectored Interrupt Controller (NVIC) in the SAM D5x/E5x family devices supports 138 interrupts with eight different priority levels. For more details, refer to the Cortex-M4 Technical Reference Manual (www.arm.com).

**Note:** Interrupts must be globally enabled using the `__enable_irq()` primitive before any interrupts will fire.

### 10.2.2. Interrupt Line Mapping

Each of the interrupt lines is connected to one peripheral instance, as shown in the table below. Each peripheral can have many interrupt flags, located in the peripheral's Interrupt Flag Status and Clear (INTFLAG) register.

An interrupt flag is set when the interrupt condition occurs. Each interrupt in the peripheral can be individually enabled by writing a '1' to the corresponding bit in the peripheral's Interrupt Enable Set (INTENSET) register, and disabled by writing '1' to the corresponding bit in the peripheral's Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated from the peripheral when the interrupt flag is set and the corresponding interrupt is enabled.

Depending on their criticality, the interrupt requests for one peripheral are either ORed together on system level, generating one interrupt or directly connected to an NVIC interrupt lines. This is described in the table below.

An interrupt request will set the corresponding interrupt pending bit in the NVIC interrupt pending registers (SETPEND/CLRPEND bits in ISPR/ICPR).

For the NVIC to activate the interrupt, it must be enabled in the NVIC interrupt enable register (SETENA/CLRENA bits in ISER/ICER). The NVIC interrupt priority registers IPR0-IPR7 provide a priority field for each interrupt.

**Table 10-1.** Interrupt Line Mapping

| Module | Source | Line |
|---|---|---|
| EIC NMI - External Interrupt Control | NMI | NMI |
| PM - Power Manager | SLEEPRDY | 0 |
| MCLK - Main Clock | CKRDY | 1 |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| OSCCTRL - Oscillators Control | XOSCFAIL0 | 2 |
| | XOSCRDY0 | |
| | XOSCFAIL1 | 3 |
| | XOSCRDY1 | |
| | DFLLLCKC | 4 |
| | DFLLLCKF | |
| | DFLLOOB | |
| | DFLLRCS | |
| | DFLLRDY | |
| | DPLL00LCKF | 5 |
| | DPLL0LCKR | |
| | DPLL0LDRTO | |
| | DPLL0LTO | |
| | DPLL1LCKF | 6 |
| | DPLL1LCKR | |
| | DPLL1LDRTO | |
| | DPLL1LTO | |
| OSC32KCTRL - 32 kHz Oscillators Control | XOSC32KFAIL | 7 |
| | XOSC32KRDY | |
| SUPC - Supply Controller | BOD33RDY | 8 |
| | B33SRDY | |
| | VCORERDY | |
| | VREGRDY | |
| | BOD33DET | 9 |
| WDT - Watchdog Timer | EW | 10 |
| RTC - Real-Time Counter | CMP0 | 11 |
| | CMP1 | |
| | CMP2 | |
| | CMP3 | |
| | OVF | |
| | PER0 | |
| | PER1 | |
| | PER2 | |
| | PER3 | |
| | PER4 | |
| | PER5 | |
| | PER6 | |
| | PER7 | |
| | TAMPER | |
| | ALARM0 | |
| | ALARM1 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| EIC - External Interrupt Controller | EXTINT 0 | 12 |
| | EXTINT 1 | 13 |
| | EXTINT 2 | 14 |
| | EXTINT 3 | 15 |
| | EXTINT 4 | 16 |
| | EXTINT 5 | 17 |
| | EXTINT 6 | 18 |
| | EXTINT 7 | 19 |
| | EXTINT 8 | 20 |
| | EXTINT 9 | 21 |
| | EXTINT 10 | 22 |
| | EXTINT 11 | 23 |
| | EXTINT 12 | 24 |
| | EXTINT 13 | 25 |
| | EXTINT 14 | 26 |
| | EXTINT 15 | 27 |
| FREQM - Frequency Meter | DONE | 28 |
| NVMCTRL - Non-Volatile Memory Controller(1) | DONE | 29 |
| | ADDRE | |
| | PROGE | |
| | LOCKE | |
| | ECCSE | |
| | ECCDE | |
| | NVME | |
| | SUSPE | |
| | SEESFULL | 30 |
| | SEESOVF | |
| | SEEWRC | |
| DMAC - Direct Memory Access Controller | SUSP 0 | 31 |
| | TCMPL 0 | |
| | TERR 0 | |
| | SUSP 1 | 32 |
| | TCMPL 1 | |
| | TERR 1 | |
| | SUSP 2 | 33 |
| | TCMPL 2 | |
| | TERR 2 | |
| | SUSP 3 | 34 |
| | TCMPL 3 | |
| | TERR 3 | |
| | SUSP 4..31 | 35 |
| | TCMPL 4..31 | |
| | TERR 4..31 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| EVSYS - Event System Interface | EVD 0 | 36 |
| | OVR 0 | |
| | EVD 1 | 37 |
| | OVR 1 | |
| | EVD 2 | 38 |
| | OVR 2 | |
| | EVD 3 | 39 |
| | OVR 3 | |
| | EVD 4..11 | 40 |
| | OVR 4..11 | |
| PAC - Peripheral Access Controller | ERR | 41 |
| RAM ECC | SINGLEE | 45 |
| | DUALE | |
| SERCOM0 - Serial Communication Interface 0(1) | 0 | 46 |
| | 1 | 47 |
| | 2 | 48 |
| | 3 | 49 |
| | 4 | |
| | 5 | |
| | 7 | |
| SERCOM1 - Serial Communication Interface 1(1) | 0 | 50 |
| | 1 | 51 |
| | 2 | 52 |
| | 3 | 53 |
| | 4 | |
| | 5 | |
| | 7 | |
| SERCOM2 - Serial Communication Interface 2(1) | 0 | 54 |
| | 1 | 55 |
| | 2 | 56 |
| | 3 | 57 |
| | 4 | |
| | 5 | |
| | 7 | |
| SERCOM3 - Serial Communication Interface 3(1) | 0 | 58 |
| | 1 | 59 |
| | 2 | 60 |
| | 3 | 61 |
| | 4 | |
| | 5 | |
| | 7 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| SERCOM4 - Serial Communication Interface 4[1] | 0 | 62 |
| | 1 | 63 |
| | 2 | 64 |
| | 3 | 65 |
| | 4 | |
| | 5 | |
| | 7 | |
| SERCOM5 - Serial Communication Interface 5[1] | 0 | 66 |
| | 1 | 67 |
| | 2 | 68 |
| | 3 | 69 |
| | 4 | |
| | 5 | |
| | 7 | |
| SERCOM6 - Serial Communication Interface 6[1] | 0 | 70 |
| | 1 | 71 |
| | 2 | 72 |
| | 3 | 73 |
| | 4 | |
| | 5 | |
| | 7 | |
| SERCOM7 - Serial Communication Interface 7[1] | 0 | 74 |
| | 1 | 75 |
| | 2 | 76 |
| | 3 | 77 |
| | 4 | |
| | 5 | |
| | 7 | |
| CAN0 - Control Area Network 0 | LINE 0 | 78 |
| | LINE 1 | |
| CAN1 - Control Area Network 1 | LINE 0 | 79 |
| | LINE 1 | |


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| USB - Universal Serial Bus | EORSM/DNRSM | 80 |
| | EORST/RST | |
| | LPM/DCONN | |
| | LPMSUSP/DDISC | |
| | RAMACER | |
| | RXSTP/TXSTP 0..7 | |
| | STALL0/STALL 0..7 | |
| | STALL1 0..7 | |
| | SUSPEND | |
| | TRFAIL0/TRFAIL 0..7 | |
| | TRFAIL1/PERR 0..7 | |
| | UPRSM | |
| | WAKEUP | |
| | SOF/HSOF | 81 |
| | TRCPT0 0..7 | 82 |
| | TRCPT1 0..7 | 83 |
| GMAC - Ethernet MAC | GMAC | 84 |
| | WOL | |
| TCC0 - Timer Counter Control 0 | CNT | 85 |
| | DFS | |
| | ERR | |
| | FAULTA | |
| | FAULTB | |
| | FAULT0 | |
| | FAULT1 | |
| | OVF | |
| | TRG | |
| | UFS | |
| | MC0 | 86 |
| | MC1 | 87 |
| | MC2 | 88 |
| | MC3 | 89 |
| | MC4 | 90 |
| | MC5 | 91 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| TCC1 - Timer Counter Control 1 | CNT | 92 |
| | DFS | |
| | ERR | |
| | FAULTA | |
| | FAULTB | |
| | FAULT0 | |
| | FAULT1 | |
| | OVF | |
| | TRG | |
| | UFS | |
| | MC0 | 93 |
| | MC1 | 94 |
| | MC2 | 95 |
| | MC3 | 96 |
| TCC2 - Timer Counter Control 2 | CNT | 97 |
| | DFS | |
| | ERR | |
| | FAULTA | |
| | FAULTB | |
| | FAULT0 | |
| | FAULT1 | |
| | OVF | |
| | TRG | |
| | UFS | |
| | MC0 | 98 |
| | MC1 | 99 |
| | MC2 | 100 |
| TCC3 - Timer Counter Control 3 | CNT | 101 |
| | DFS | |
| | ERR | |
| | FAULTA | |
| | FAULTB | |
| | FAULT0 | |
| | FAULT1 | |
| | OVF | |
| | TRG | |
| | UFS | |
| | MC0 | 102 |
| | MC1 | 103 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| TCC4 - Timer Counter Control 4 | CNT | 104 |
| | DFS | |
| | ERR | |
| | FAULTA | |
| | FAULTB | |
| | FAULT0 | |
| | FAULT1 | |
| | OVF | |
| | TRG | |
| | UFS | |
| | MC0 | 105 |
| | MC1 | 106 |
| TC0 - Basic Timer Counter 0 | ERR | 107 |
| | MC0 | |
| | MC1 | |
| | OVF | |
| TC1 - Basic Timer Counter 1 | ERR | 108 |
| | MC0 | |
| | MC1 | |
| | OVF | |
| TC2 - Basic Timer Counter 2 | ERR | 109 |
| | MC0 | |
| | MC1 | |
| | OVF | |
| TC3 - Basic Timer Counter 3 | ERR | 110 |
| | MC0 | |
| | MC1 | |
| | OVF | |
| TC4 - Basic Timer Counter 4 | ERR | 111 |
| | MC0 | |
| | MC1 | |
| | OVF | |
| TC5 - Basic Timer Counter 5 | ERR | 112 |
| | MC0 | |
| | MC1 | |
| | OVF | |
| TC6 - Basic Timer Counter 6 | ERR | 113 |
| | MC0 | |
| | MC1 | |
| | OVF | |
| TC7 - Basic Timer Counter 7 | ERR | 114 |
| | MC0 | |
| | MC1 | |
| | OVF | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| PDEC - Position Decoder | DIR | 115 |
| | ERR | |
| | OVF | |
| | VLC | |
| | MC0 | 116 |
| | MC1 | 117 |
| ADC0 - Analog Digital Converter 0 | OVERRUN | 118 |
| | WINMON | |
| | RESRDY | 119 |
| ADC1 - Analog Digital Converter 1 | OVERRUN | 120 |
| | WINMON | |
| | RESRDY | 121 |
| AC - Analog Comparators | COMP0 | 122 |
| | COMP1 | |
| | WIN0 | |
| DAC - Digital-to-Analog Converter | OVERRUN0 | 123 |
| | OVERRUN1 | |
| | UNDERRUN0 | |
| | UNDERRUN1 | |
| | EMPTY0 | 124 |
| | EMPTY1 | 125 |
| | RESRDY0 | 126 |
| | RESRDY1 | 127 |
| I2S - Inter-IC Sound Interface | RXOR0 | 128 |
| | RXOR1 | |
| | RXRDY0 | |
| | RXRDY1 | |
| | TXRDY0 | |
| | TXRDY1 | |
| | TXUR0 | |
| | TXUR1 | |
| PCC - Parallel Capture Controller | DRDY | 129 |
| | OVRE | |
| AES - Advanced Encryption Standard | ENCCMP | 130 |
| | GFMCMP | |
| TRNG - True Random Generator | DATARDY | 131 |
| ICM - Integrity Check Monitor | All ICM Interrupts | 132 |
| Reserved | Reserved | 133 |
| QSPI - Quad SPI interface | RXC | 134 |
| | DRE | |
| | TXC | |
| | ERROR | |
| | CSRISE | |
| | INSTREND | |
| SDHC0 - SD/MMC Host Controller 0 | All SDHC0 Interrupts | 135 |


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

**Table 10-1.** Interrupt Line Mapping (continued)

| Module | Source | Line |
|---|---|---|
| SDHC1 - SD/MMC Host Controller 1 | All SDHC1 Interrupts | 136 |

**Note:**

1. The integer number specified in the source refers to the respective bit position in the INTFLAG register of respective peripheral.

**Note:** Lines not listed here are reserved.

## 10.3. High-Speed Bus System

### 10.3.1. Features

High-Speed Bus Matrix has the following features:

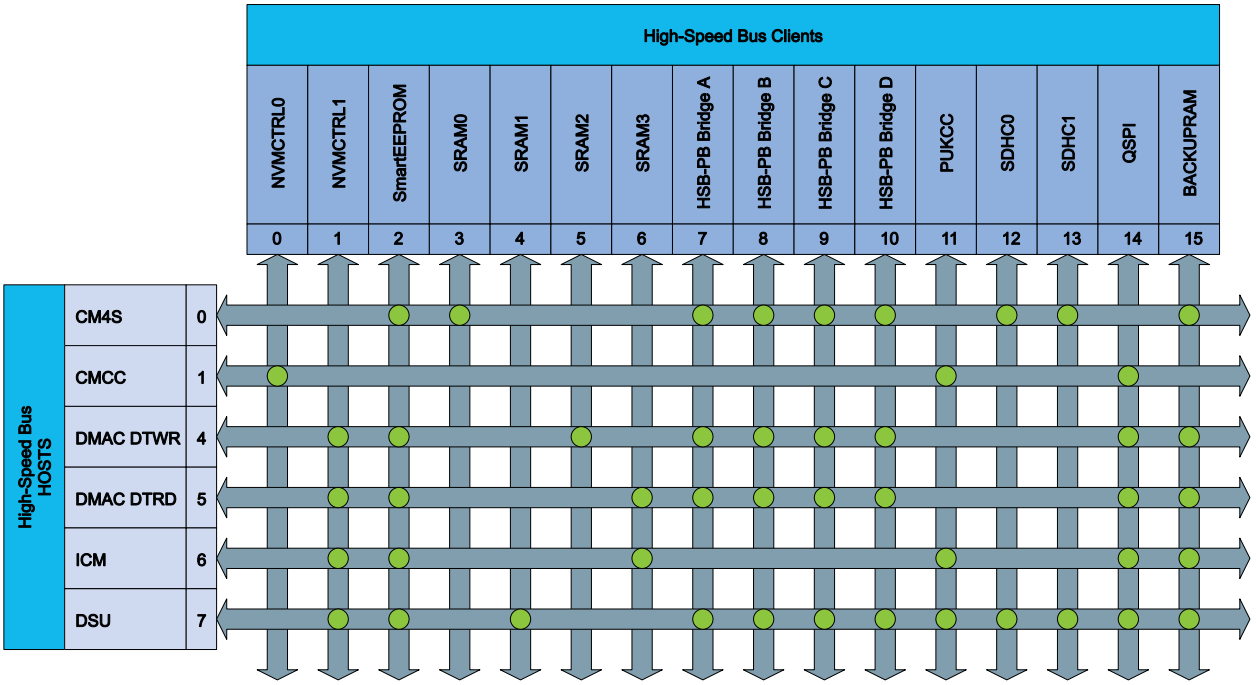
- Symmetric crossbar bus switch implementation
- Allows concurrent accesses from different Hosts to different Clients
- 32-bit data bus
- Operation at a one-to-one clock frequency with the bus Hosts

FlexRAM Memory has the following features:

- Unified System Memory area
- Allows concurrent accesses from different Hosts
- Offers privileged accesses from specific Hosts

### 10.3.2. Configuration

**Figure 10-1.** Host-Client Relations High-Speed Bus Matrix





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 10-2.** High-Speed Bus Matrix Hosts

| High-Speed Bus Matrix Hosts | Host ID |
|---|---|
| CM4S - Cortex M4 Processor | 0 |
| CMCC - Cortex-M Cache Controller (Cortex-M4F ICode and DCode Interfaces) | 1 |
| DMAC - Direct Memory Access Controller/Data Write Access | 4 |
| DMAC - Direct Memory Access Controller/Data Read Access | 5 |
| ICM - Integrity Check Monitor | 6 |
| DSU - Device Service Unit | 7 |

**Table 10-3.** High-Speed Bus Matrix Clients

| High-Speed Bus Matrix Clients | Client ID |
|---|---|
| Internal Flash Memory | 0, 1 |
| SmartEEPROM | 2 |
| SRAM Port 0 - CM4 Access | 3 |
| SRAM Port 1 - DSU Access | 4 |
| SRAM Port 2 - DMAC Data-Write Access | 5 |
| SRAM Port 3 - DMAC Data-Read and ICM Access | 6 |
| AHB-APB Bridge A | 7 |
| AHB-APB Bridge B | 8 |
| AHB-APB Bridge C | 9 |
| AHB-APB Bridge D | 10 |
| PUKCC | 11 |
| SDHC0 | 12 |
| SDHC1 | 13 |
| QSPI | 14 |
| BACKUP RAM Memory | 15 |

### 10.3.3. SRAM Quality of Service

To ensure that Hosts with latency requirements get sufficient priority when accessing RAM, priority levels can be assigned to the Hosts for different types of access.

The Quality of Service (QoS) level is independently selected for each Host accessing the RAM. For any access to the RAM, the RAM also receives the QoS level. The QoS levels and their corresponding bit values for the QoS level configuration is shown in the table below.

**Table 10-4.** Quality of Service

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Background (no sensitive operation) |
| 0x1 | LOW | Sensitive Bandwidth |
| 0x2 | MEDIUM | Sensitive Latency |
| 0x3 | HIGH | Critical Latency |

If a Host is configured with QoS level DISABLE (0x0) or LOW (0x1) there will be a minimum latency of one cycle for the RAM access.

The priority order for concurrent accesses are decided by two factors. First, the QoS level for the Host and second, a static priority given by the port ID. The lowest port ID has the highest static priority. See the tables below for details.

The CPU QoS level can be written/read, using 32-bit access only, at address 0x4100C11C, bits [1:0]. Its reset value is 0x3.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The ICM QoS level can be written/read, using 32-bit access only, at address 0x4100C128, bits [1:0]. Its reset value is 0x1.

Refer to different Host QOS control registers for configuring QoS for the other Hosts (DSU, DMAC, CAN, USB).

**Table 10-5.** SRAM Port Connections QoS

| SRAM Port Connection | Port ID | Connection Type | QoS | default QoS |
|---|---|---|---|---|
| CM4 - Cortex M4 Processor | 0 | Bus Matrix | 0x4100C11C, bits[1:0](1) | 0x3 |
| DSU - Device Service Unit | 1 | Bus Matrix | IP-CFG.LQOS | 0x2 |
| DMAC - Direct Memory Access Controller - Data Access | 2 (WR), 3 (RD) | Bus Matrix | IP-PRICTRL0.QOSn | 0x2 |
| ICM - Integrity Check Monitor | 3 | Bus Matrix | 0x4100C128, bits[1:0](1) | 0x1 |
| DMAC - Direct Memory Access Controller - Fetch Access | 4, 5 | Direct | IP-PRICTRL0.QOSn | 0x2 |
| DMAC - Direct Memory Access Controller - Write-Back Access | 6, 7 | Direct | IP-PRICTRL0.QOSn | 0x2 |
| SDHC0 - SD/MMC Host Controller | 8 | Direct | STATIC-1 | 0x1 |
| SDHC1 - SD/MMC Host Controller | 9 | Direct | STATIC-1 | 0x1 |
| CAN0 - Control Area Network | 10 | Direct | IP-MRCFG.QOS | 0x1 |
| CAN1 - Control Area Network | 11 | Direct | IP-MRCFG.QOS | 0x1 |
| GMAC - Ethernet MAC | 12 | Direct | STATIC-2 | 0x2 |
| USB - Universal Serial Bus - Configuration Access | 13 | Direct | IP-QOSCTRL.CQOS | 0x3 |
| USB - Universal Serial Bus - Data Access | 13 | Direct | IP-QOSCTRL.DQOS | 0x3 |

**Note:** 1. Using 32-bit access only.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 11. CMCC - Cortex M Cache Controller

## 11.1. Overview

The Cortex M Cache Controller provides an L1 cache to the Cortex M CPU. The CMCC sits transparently between the CPU and the cache leading to improved performance.

The CMCC interfaces with the CPU through the AHB and is connected to the APB bus interface for its configuration.

## 11.2. Features

- Physically addressed and physically tagged
- L1 data and instruction cache set to 4 KB
- L1 cache line size set to 16 Bytes
- L1 cache integrates 32-bit bus Host interface
- Unified 4-Way set associative cache architecture
- Lock-Down feature, which allows cached to be locked per way
- Write through cache operations, read allocate
- Configurable as data and instruction Tightly Coupled Memory (TCM)
- Round Robin victim selection policy
- Event Monitoring, with one programmable 32-bit counter
- Cache Interface includes cache maintenance operations registers


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 11.3. Block Diagram

**Figure 11-1.** CMCC Block Diagram

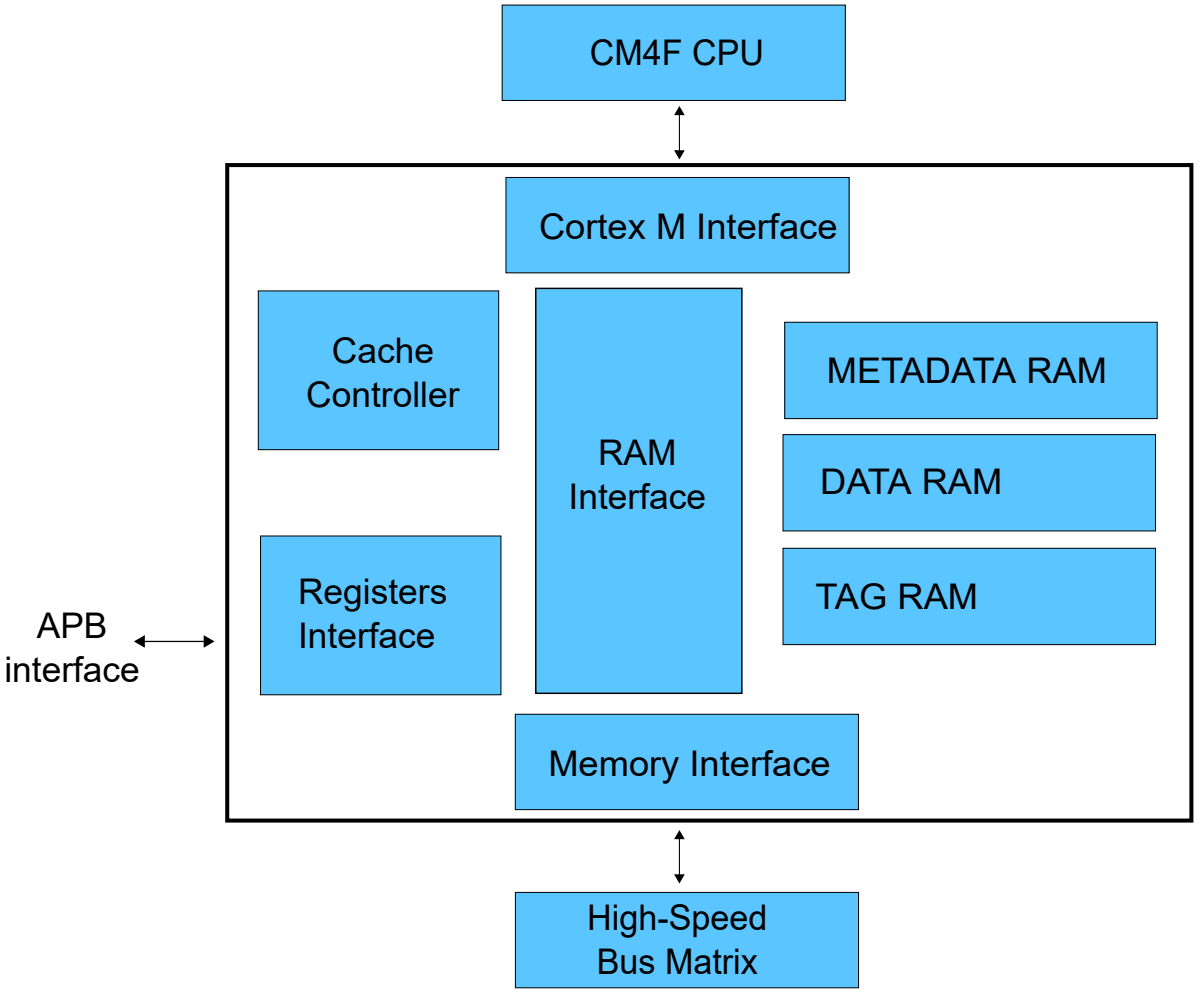
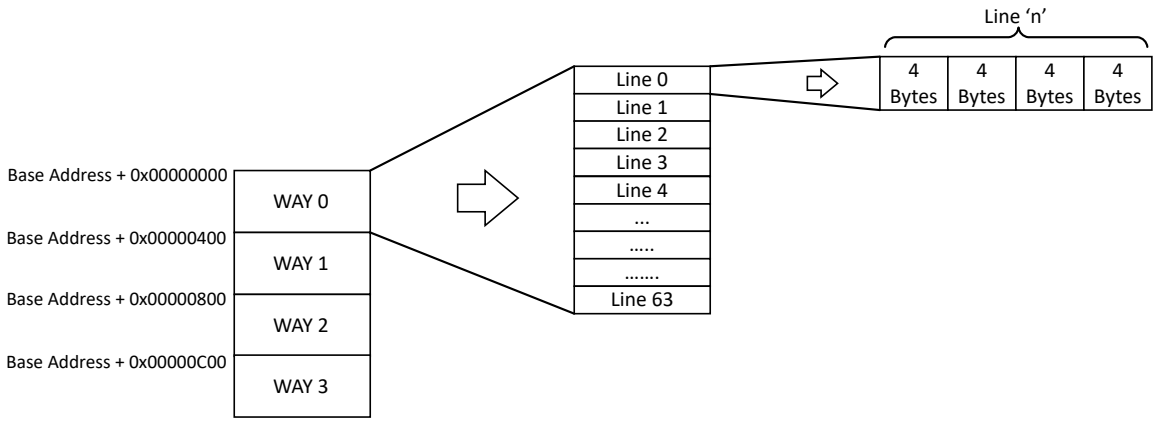



**Figure 11-2.** CMCC Organization





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 11.4. Signal Description

Not applicable.

## 11.5. Product Dependencies

Not applicable.

### 11.5.1. I/O Lines

Not applicable.

### 11.5.2. Power Management

The CMCC will continue to function as long as the CPU is not sleeping and the CMCC is enabled.

### 11.5.3. Clocks

Not applicable.

### 11.5.4. DMA

Not applicable.

### 11.5.5. Interrupts

Not applicable.

### 11.5.6. Events

Not applicable.

### 11.5.7. Debug Operation

When the CPU is halted in debug mode, the CMCC is halted. Any read access by the debugger in cached zones are not cached.

### 11.5.8. Register Access Protection

Not applicable.

### 11.5.9. Analog Connections

Not applicable.

## 11.6. Functional Description

### 11.6.1. Principle of Operation

### 11.6.2. Initialization and Normal Operation

On reset, the cache controller data entries are all invalidated, and the cache is disabled. The cache is transparent to processor operations. The cache controller is activated through the use of its configuration registers. The configuration interface is memory mapped in the APB bus.

Use the following sequence to enable the cache controller:

- Verify that the CMCC is disabled, reading the value of the SR.CSTS.
- Enable the CMCC by writing '1' in CTRL.CEN. The MODULE is disabled by writing a '0' in CTRL.CEN.

### 11.6.3. Change Cache Size

It is possible to change the cache size by writing to the Cache Size Configured By Software bits in the Cache Configuration register (CFG.CSIZESW).

Use the following sequence to change the cache size:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Disable the CMCC controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN=0).
- Check the Cache Controller Status bit in the Cache Status register to verify that the CMCC is successfully disabled (SR.CSTS=0).
- Change CFG.CSIZESW to its new value.
- Enable the CMCC by writing CTRL.CEN=1.

### 11.6.4. Data Cache Disable

The Instructions alone can be cached by disabling the Data cache, as described in the following steps:

1. Disable the cache controller by writing a '0' to CTRL.CEN.
2. Check SR.CSTS to verify whether the CMCC is successfully disabled.
3. Write CFG.DCDIS = 1.
4. Enable the CMCC by writing CTRL.CEN = 1.

### 11.6.5. Instruction Cache Disable

The Data alone can be cached by disabling the Instruction cache, as described in the following steps:

1. Disable the cache controller by writing CTRL.CEN = 0.
2. Check SR.CSTS to verify that the CMCC is successfully disabled.
3. Write CFG.ICDIS = 1.
4. Enable the CMCC by writing CTRL.CEN = 1.

### 11.6.6. Cache Load and Lock

It is possible to lock a specific way for code optimization by writing the Lock Way register (LCKWAY.LCKWAY). The locked way will not be updated by the CMCC as part of cache operations.

The load and lock mechanism can be implemented to use cache memory in a deterministic way. Follow these steps to load and lock a way:

1. Disable cache controller by clearing the CTRL.CEN bit.
2. Invalidate the desired WAY line by line. This will reset the round robin algorithm of the invalidated line, that will become eligible for the next load operation.
3. Disable the Instruction cache, but keep the Data cache enabled.
4. Enable the cache by setting the CTRL.CEN bit.
5. Place the respective piece of code and/or data to the corresponding WAY due to simple LOAD operations. Loading the piece of code and/or data will force the cache to refill the previous invalidated line in the right way. No need to load all the bytes of the line, only the first byte. The cache will automatically refill the complete line.
6. Lock the specific WAY by setting LCKWAY.LCKWAY[3:0].
7. Re-enable the instruction cache. The locked WAY is now loaded and ready to operate. The remaining WAYS can be used as I-cache or D-cache as required.

### 11.6.7. Tightly Coupled Memory

Users can use a part of the cache as Tightly Coupled Memory (TCM). The cache size is determined by the Cache Size Configuration by Software bits in the Cache Configuration register (CFG.CSIZESW). The relation between cache and TCM is as given below:

TCM size = maximum Cache size - configured Cache size.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The TCM start address can be obtained from the product memory mapping. The cache memory starts first from the address followed by the TCM memory. Size of the Way is fixed and the number of ways varies according to the available size for the cache memory. For additional information, refer to the section Product Memory Mapping.

**Table 11-1.** TCM Sizes

| Max. Cache | Configured Cache | TCM Size |
|---|---|---|
| 4 KB | 4 KB | 0 KB |
| 4 KB | 1 KB | 3 KB |
| 4 KB | 2 KB | 2 KB |
| 4 KB | 0 KB | 4 KB |

The TCM is also accessible in its maximum size when the CMCC is disabled. The TCM does not need to be locked in order to operate.

**Note:** Writing into the cache DATA RAM region through the CPU can overwrite the valid cache lines. This can result in data corruption when the cache controller is accessing the data for cache transactions. Access the DATA RAM region only after configuring it as TCM.

### 11.6.8. Cache Maintenance

#### 11.6.8.1. Cache Invalidate by Line Operation

When an invalidate by line command is issued, the CMCC resets the valid bit information of the decoded cache line. As the line is no longer valid, the replacement counter points to that line.

- Disable the cache controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN).
- Check SR.CSTS to verify that the CMCC is successfully disabled.
- Perform an invalidate by line by writing the set {index,way} in the Cache Maintenance 1 register (MAINT1.INDEX, MAINT1.WAY).
- Enable the CMCC by writing a '1' to CTRL.CEN.

#### 11.6.8.2. Cache Invalidate All Operation

Use the following sequence to invalidate all cache entries.

- Disable the cache controller by writing a zero to the Cache Enable bit in the Cache Control register (CTRL.CEN).
- Check SR.CSTS to verify that the CMCC is successfully disabled.
- Perform a full invalidate operation by writing a '1' to the Cache Controller Invalidate All bit in the Cache Maintenance 0 register (MAINT0.INVALL).
- Enable the CMCC by writing a '1' to CTRL.CEN.

### 11.6.9. Cache Performance Monitoring

The Cortex M cache controller includes a programmable monitor/32-bit counter. The monitor can be configured to count the number of clock cycles, the number of data hit or the number of instruction hit.

It is important to know that the Cortex-M4 processor prefetches instructions ahead of execution. It performs only 32-bit read access on the Instruction Bus, which means:

- One arm instruction is fetched per bus access
- Two thumb instructions are fetched per bus access

As a consequence, two thumb instructions (e.g., `NOP`) need one bus access, which results in the HIT counter incrementing by 1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Use the following sequence to activate the counter:

- Configure the monitor counter by writing the MCFG.MODE.
  - CYCLE_COUNT is used to increment the counter along with the program counter, to count the number of cycles.
  - IHIT_COUNT is the instruction Hit counter, which increments the counter when there is a hit for the instruction in the cache.
  - DHIT_COUNT is the data Hit counter which increments the counter when there is a hit for the data in the cache.
- Enable the counter by writing a '1' to the Cache Controller Monitor Enable bit in the Cache Monitor Enable register (MEN.MENABLE).
- If required, reset the counter by writing a '1' to the Cache Controller Software Reset bit in the Cache Monitor Control register (MCTRL.SWRST).
- Check the value of the monitor counter by reading the MSR.EVENT_CNT bit field.

## 11.7. DEBUG Mode

In Debug mode, TAG and METADATA RAM blocks content is read/written through the AHB bus interface if the CMCC is disabled. When the CMCC is enabled, the TAG and METADATA RAM blocks are non readable.

Debug access has the same R/W properties as the CPU access for the DATA RAM block.

The TAG, METADATA and DATA RAM blocks' R/W properties are summarized in RAM Properties.

Use the following sequence to perform read access with the Debugger to the three RAM blocks:

- Disable the cache controller by writing a zero to the Cache Controller Enable bit in the Cache Control register (CTRL.CEN).
- Check the Cache Controller Status bit in the Cache Status register (SR.CSTS) to verify that the CMCC is successfully disabled.
- Perform a read or write access through Debugger:
  - @ CMCC_AHB_ADDR for DATA RAM,
  - @ CMCC_AHB_ADDR_TAG for TAG RAM,
  - @ CMCC_AHB_ADDR_MTDATA for METADATA RAM.
- If a write access has been performed in the TAG, METADATA, or DATA RAM in the cache section, an invalid operation must be performed before re-enabling the CMCC.

**Related Links**
RAM Properties

## 11.8. RAM Properties

The following table shows the different access properties of the three RAM blocks, according the different modes described in the previous chapters.

**Table 11-2.** Access to RAM

| Access Condition | DATA RAM | TAG RAM | METADATARAM |
|---|---|---|---|
| CPU access when CMCC DISABLED | R/W | no R/W - hardfault | no R/W - hardfault |
| CPU access when CMCC ENABLED | CACHE section configured: R/W(1)<br>TCM section configured: R/W | no R/W - hardfault | no R/W - hardfault |
| Debugger access when CMCC DISABLED | R/W | R/W | R/W |
| Debugger access when CMCC ENABLED | CACHE section configured: R/W(1)<br>TCM section configured: R/W | no R/W | no R/W |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:**

1. A write operation in this zone can corrupt the coherency of the cache. An invalidate operation may be needed.

**Related Links**

DEBUG Mode


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

## 11.9. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | TYPE | 7:0 | LCKDOWN | WAYNUM[1:0] | | RRP | | | GCLK | |
| | | 15:8 | | | CLSIZE[2:0] | | | CSIZE[2:0] | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CFG | 7:0 | | CSIZESW[2:0] | | | | DCDIS | ICDIS | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRL | 7:0 | | | | | | | | CEN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | SR | 7:0 | | | | | | | | CSTS |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | LCKWAY | 7:0 | | | | | LCKWAY[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | MAINT0 | 7:0 | | | | | | | | INVALL |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | MAINT1 | 7:0 | INDEX[3:0] | | | | | | | |
| | | 15:8 | | | | | INDEX[7:4] | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | WAY[3:0] | | | | | | | |
| 0x28 | MCFG | 7:0 | | | | | | | MODE[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x2C | MEN | 7:0 | | | | | | | | MENABLE |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x30 | MCTRL | 7:0 | | | | | | | | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | MSR | 7:0 | EVENT_CNT[7:0] | | | | | | | |
| | | 15:8 | EVENT_CNT[15:8] | | | | | | | |
| | | 23:16 | EVENT_CNT[23:16] | | | | | | | |
| | | 31:24 | EVENT_CNT[31:24] | | | | | | | |

## 11.10. Register Description


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.1. Cache Type

**Name:** TYPE
**Offset:** 0x00
**Reset:** 0x000012D2
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | CLSIZE[2:0] | | | CSIZE[2:0] | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 1 | 0 | 0 | 1 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LCKDOWN | WAYNUM[1:0] | | RRP | | | GCLK | |
| Access | R | R | R | R | | | R | |
| Reset | 1 | 1 | 0 | 1 | | | 1 | |

**Bits 13:11 – CLSIZE[2:0]** Cache Line Size
This field configures the Cache Line Size.

| Value | Name | Description |
|---|---|---|
| 0x2 | CLSIZE_16B | Cache Line Size is 16 bytes |
| 0x3-0x7 | - | Reserved |

**Bits 10:8 – CSIZE[2:0]** Cache Size
This bit field configures the cache size.

| Value | Name | Description |
|---|---|---|
| 0x0 | CSIZE_1KB | Cache Size is 1 KB |
| 0x1 | CSIZE_2KB | Cache Size is 2 KB |
| 0x2 | CSIZE_4KB | Cache Size is 4 KB |
| 0x3-0x7 | - | Reserved |

**Bit 7 – LCKDOWN** Lock Down Supported
Writing a '0' to this bit disables the Lock Down feature.
Writing a '1' to this bit enables the Lock Down feature.

| Value | Description |
|---|---|
| 0 | Lock Down feature is not supported. |
| 1 | Lock Down feature is supported. |

**Bits 6:5 – WAYNUM[1:0]** Number of Way
This bit field configures the mapping of the cache.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x0 | DMAPPED | Direct Mapped Cache |
| 0x1 | ARCH2WAY | 2-WAY set associative |
| 0x2 | ARCH4WAY | 4-WAY set associative |
| 0x3 | - | Reserved |

**Bit 4 – RRP** Round Robin Policy Supported
Writing a '0' to this bit disables Round Robin Policy.
Writing a '1' to this bit enables Round Robin Policy.

| Value | Description |
|---|---|
| 0 | Round Robin Policy is disabled. |
| 1 | Round Robin Policy is enabled. |

**Bit 1 – GCLK** Dynamic Clock Gating
Writing a '0' to this bit disables the Dynamic Clock Gating feature.
Writing a '1' to this bit enables the Dynamic Clock Gating feature.

| Value | Description |
|---|---|
| 0 | Dynamic Clock Gating is disabled. |
| 1 | Dynamic Clock Gating is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.2. Cache Configuration

**Name:** CFG
**Offset:** 0x04
**Reset:** 0x00000020
**Property:** R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | CSIZESW[2:0] | | | | DCDIS | ICDIS | |
| Access | | R/W | R/W | R/W | | R/W | R/W | |
| Reset | | 0 | 1 | 0 | | 0 | 0 | |

**Bits 6:4 – CSIZESW[2:0]** Cache Size Configured by Software
This field configures the cache size.

| Value | Name | Description |
|---|---|---|
| 0x0 | CONF_CSIZE_1KB | The Cache Size is configured to 1KB |
| 0x1 | CONF_CSIZE_2KB | The Cache Size is configured to 2KB |
| 0x2 | CONF_CSIZE_4KB | The Cache Size is configured to 4KB |
| 0x3-0x7 | - | Reserved |

**Bit 2 – DCDIS** Data Cache Disable
Writing a '0' to this bit enables data caching.
Writing a '1' to this bit disables data caching.

| Value | Description |
|---|---|
| 0 | Data caching is enabled. |
| 1 | Data caching is disabled. |

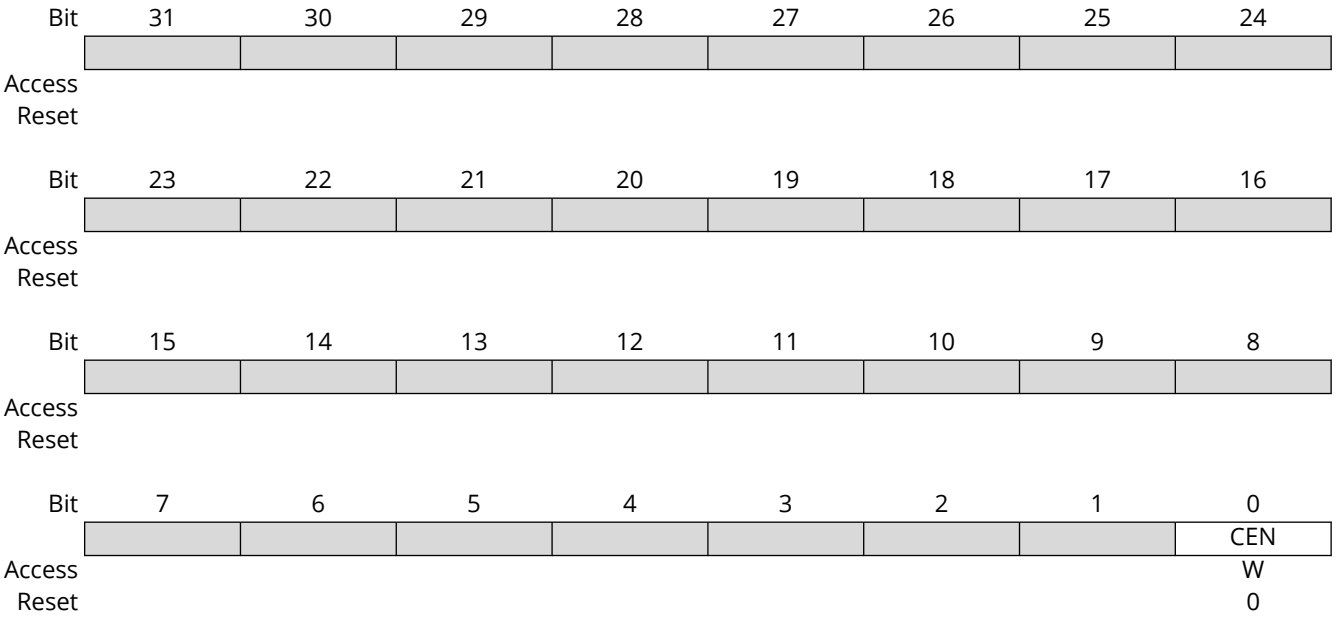
**Bit 1 – ICDIS** Instruction Cache Disable
Writing a '0' to this bit enables instruction caching.
Writing a '1' to this bit disables instruction caching.

| Value | Description |
|---|---|
| 0 | Instruction caching is enabled. |
| 1 | Instruction caching is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.3. Cache Control

**Name:** CTRL
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** Write-only




**Bit 0 – CEN** Cache Controller Enable
Writing a '0' to this bit disables the CMCC.
Writing a '1' to this bit enables the CMCC.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.4. Cache Status

**Name:** SR
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CSTS |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

**Bit 0 – CSTS** Cache Controller Status
Writing to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.5. Cache Lock per Way

**Name:** LCKWAY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LCKWAY[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

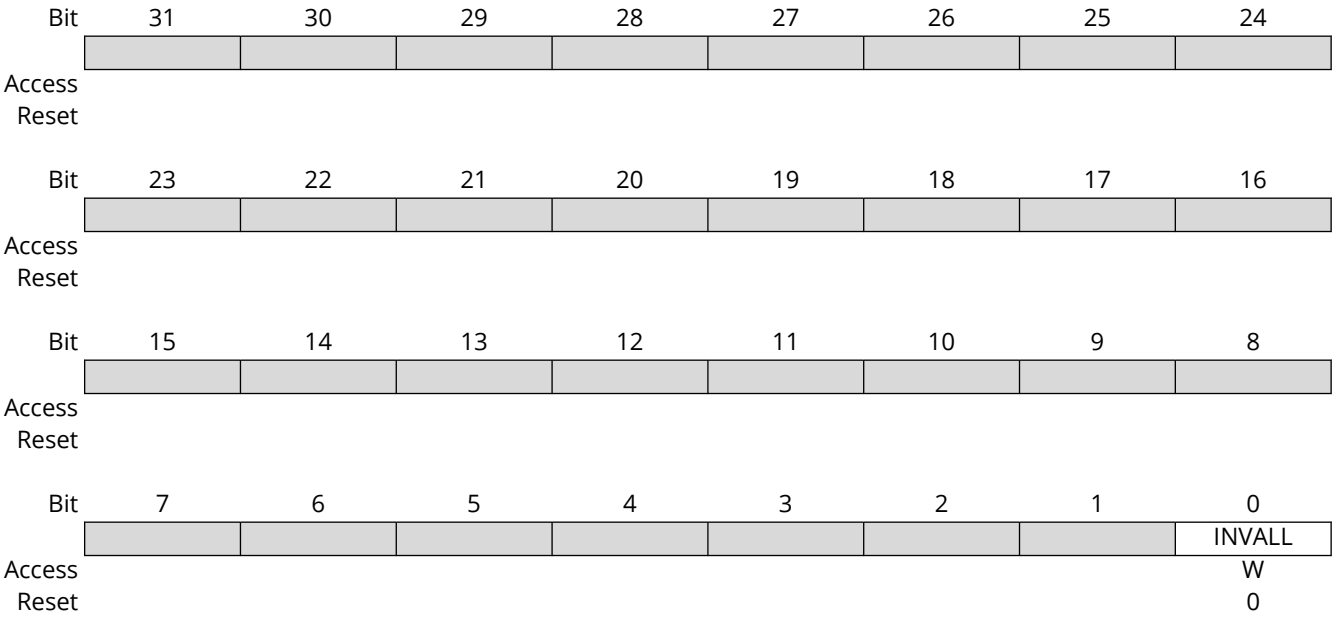
**Bits 3:0 – LCKWAY[3:0]** Lockdown Way Register
This field selects which way is locked.

| Value | Name | Description |
|---|---|---|
| 0x1 | WAY0 | WAY0 is locked |
| 0x2 | WAY1 | WAY1 is locked |
| 0x4 | WAY2 | WAY2 is locked |
| 0x8 | WAY3 | WAY3 is locked |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.6. Cache Maintenance 0

**Name:** MAINT0
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** Write-only

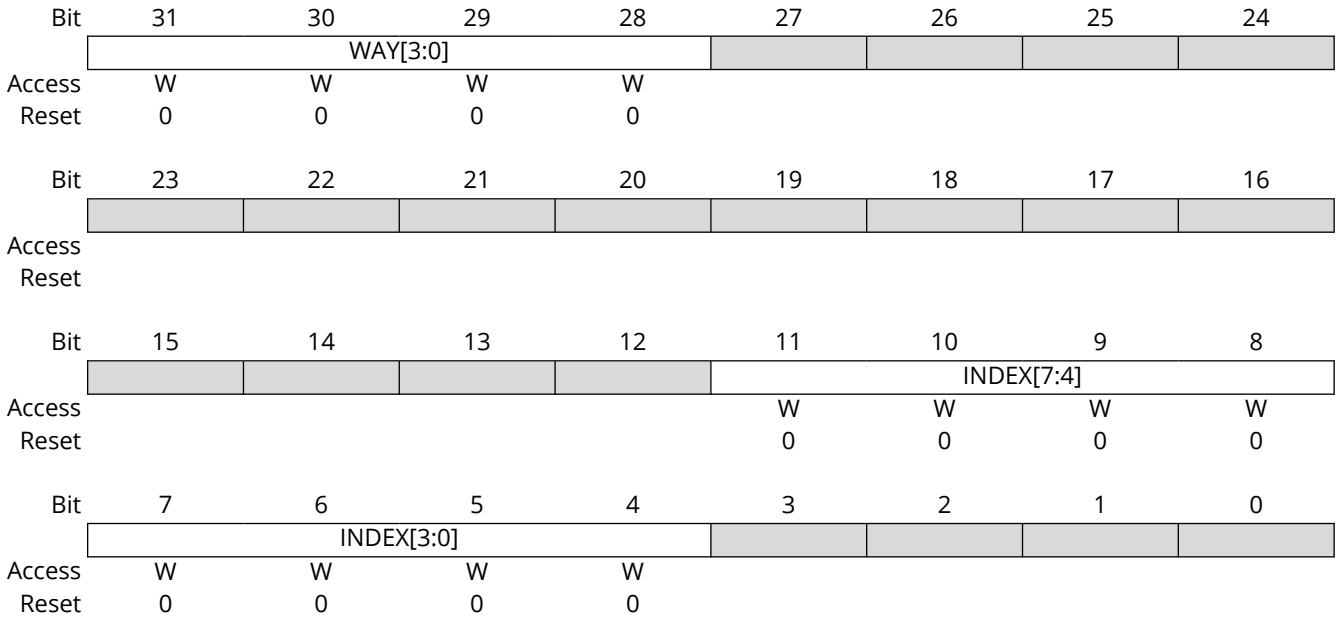



**Bit 0 – INVALL** Cache Controller Invalidate All
Writing a '0' to this bit has no effect.
Writing a '1' to this bit invalidates all cache entries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.7. Cache Maintenance 1

**Name:** MAINT1
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** Write-only




**Bits 31:28 – WAY[3:0]** Invalidate Way

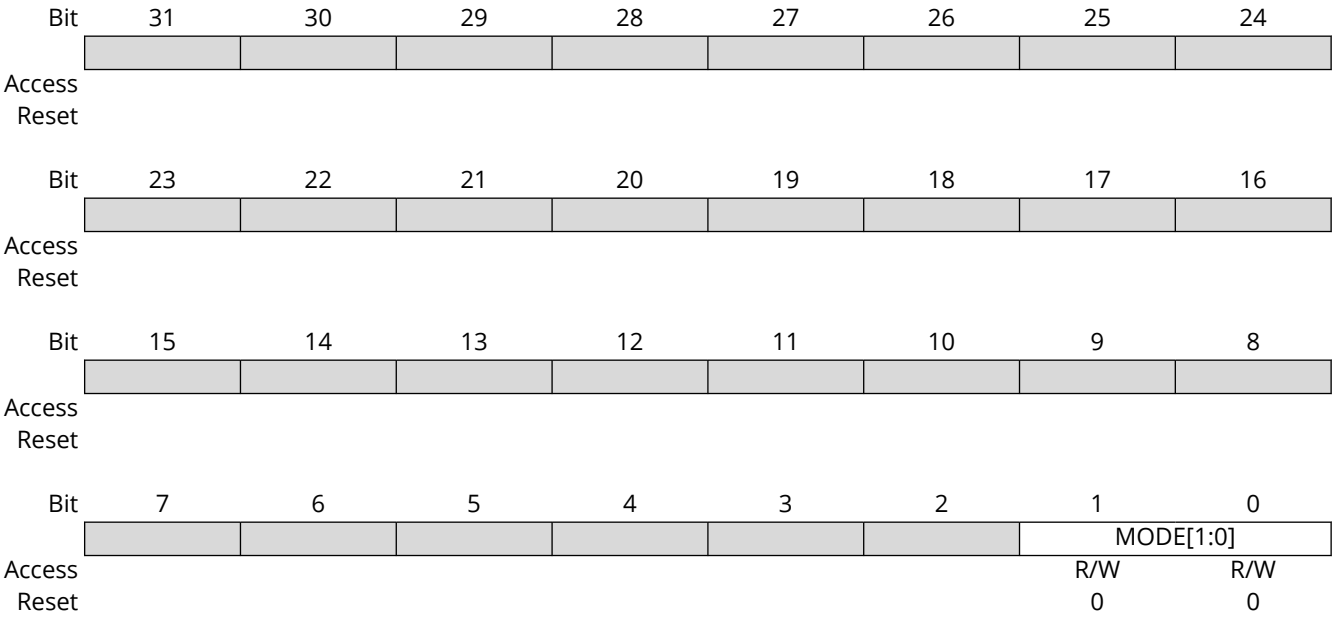
| Value | Name | Description |
|---|---|---|
| 0x1 | WAY0 | Way 0 is selection for index invalidation |
| 0x2 | WAY1 | Way 1 is selection for index invalidation |
| 0x4 | WAY2 | Way 2 is selection for index invalidation |
| 0x8 | WAY3 | Way 3 is selection for index invalidation |

**Bits 11:4 – INDEX[7:0]** Invalidate Index
This field selects the index value for invalidation


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.8. Cache Monitor Configuration

**Name:** MCFG
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** Read/Write




**Bits 1:0 – MODE[1:0]** Cache Controller Monitor Counter Mode
This field selects the type of data monitored.

| Value | Name | Description |
|---|---|---|
| 0x0 | CYCLE_COUNT | Cycle counter |
| 0x1 | IHIT_COUNT | Instruction hit counter |
| 0x2 | DHIT_COUNT | Data hit counter |
| 0x3 | | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 11.10.9. Cache Monitor Enable

**Name:** MEN
**Offset:** 0x2C
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | MENABLE |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – MENABLE** Cache Controller Monitor Enable
Writing a '0' to this bit disables the monitor counter.
Writing a '1' to this bit enables the monitor counter.

| Value | Description |
|---|---|
| 0 | The Monitor counter is disabled. |
| 1 | The Monitor counter is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.10. Cache Monitor Control

**Name:** MCTRL
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** Write-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SWRST |
| Access | | | | | | | | W |
| Reset | | | | | | | | 0 |

**Bit 0 – SWRST** Cache Controller Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the event counter register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 11.10.11. Cache Monitor Status

**Name:** MSR
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | EVENT_CNT[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | EVENT_CNT[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EVENT_CNT[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EVENT_CNT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – EVENT_CNT[31:0]** Monitor Event Counter
This field indicates the Monitor Event Counter value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 12. DSU - Device Service Unit

## 12.1. Overview

The Device Service Unit (DSU) provides a means of detecting debugger probes. It enables the ARM Debug Access Port (DAP) to have control over multiplexed debug pads and CPU Reset. The DSU also provides system-level services to debug adapters in an ARM debug system. It implements a CoreSight Debug ROM that provides device identification as well as identification of other debug components within the system. Hence, it complies with the ARM Peripheral Identification specification. The DSU also provides system services to applications that need memory testing, as required for IEC60730 Class B compliance, for example. The DSU can be accessed simultaneously by a debugger and the CPU, as it is connected on the High-Speed Bus Matrix. For security reasons, some of the DSU features will be limited or unavailable when the device is protected by the NVMCTRL security bit.

**Related Links**

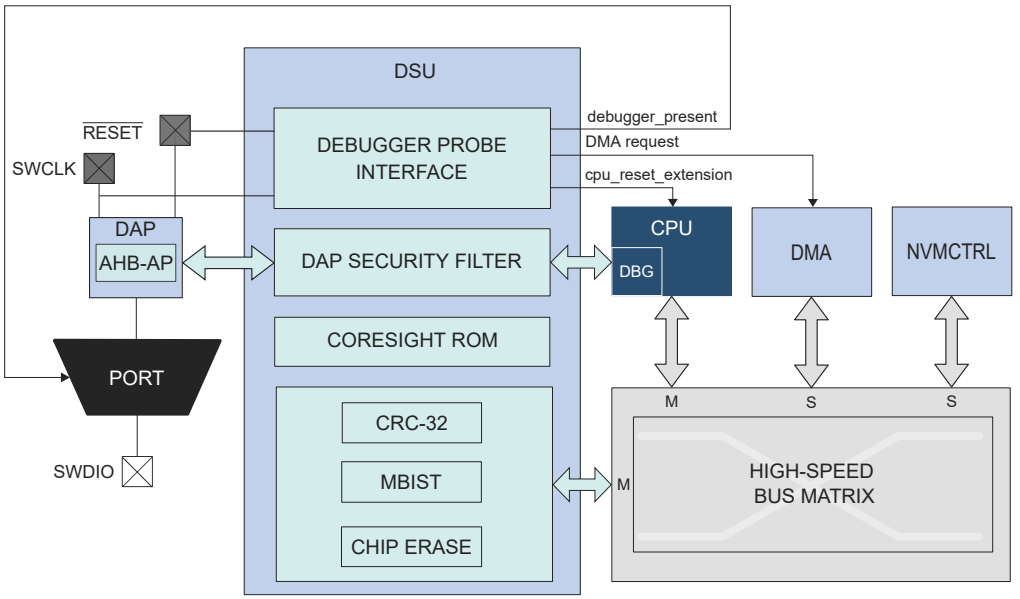
NVMCTRL – Nonvolatile Memory Controller

## 12.2. Features

- CPU Reset Extension
- Debugger Probe Detection (Cold- and Hot-Plugging)
- Chip-Erase Command and Status
- 32-Bit Cyclic Redundancy Check (CRC32) of any Memory Accessible through the Bus Matrix
- ARM® CoreSight™ Compliant Device Identification
- Two Debug Communications Channels with DMA Connection
- Debug Access Port Security Filter
- Onboard Memory Built-in Self-test (MBIST)

## 12.3. Block Diagram

**Figure 12-1.** DSU Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 12.4. Signal Description

The DSU uses three signals to function.

| Signal Name | Type | Description |
|---|---|---|
| $\overline{\text{RESET}}$ | Digital Input | External Reset (RESETN pin) |
| SWCLK | Digital Input | SW clock pin |
| SWDIO | Digital I/O | SW bidirectional data pin |

**Related Links**

I/O Multiplexing and Considerations

## 12.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 12.5.1. I/O Lines

The SWCLK pin is by default assigned to the DSU module to allow debugger probe detection and to stretch the CPU Reset phase. For more information, refer to Debugger Probe Detection. The Hot-Plugging feature depends on the PORT configuration. If the SWCLK pin function is changed in the port or if the PORT_MUX is disabled, the Hot-Plugging feature is disabled until a power reset or an external Reset is performed.

### 12.5.2. Power Management

The DSU will continue to operate in Idle mode.

**Related Links**

PM – Power Manager

### 12.5.3. Clocks

The DSU bus clocks (CLK_DSU_APB and CLK_DSU_AHB) can be enabled and disabled by the Main Clock Controller.

**Related Links**

PM – Power Manager
MCLK – Main Clock
Peripheral Clock Masking

### 12.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to DMAC – Direct Memory Access Controller for details.

**Related Links**

DMAC – Direct Memory Access Controller

### 12.5.5. Interrupts

Not applicable.

### 12.5.6. Events

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.5.7. Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Debug Communication Channel 0 register (DCC0)
- Debug Communication Channel 1 register (DCC1)

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

**Related Links**

PAC - Peripheral Access Controller

### 12.5.8. Analog Connections

Not applicable.

## 12.6. Debug Operation

### 12.6.1. Principle of Operation

The DSU provides basic services to allow on-chip debug using the ARM Debug Access Port and the ARM processor debug resources:

- CPU Reset extension
- Debugger probe detection

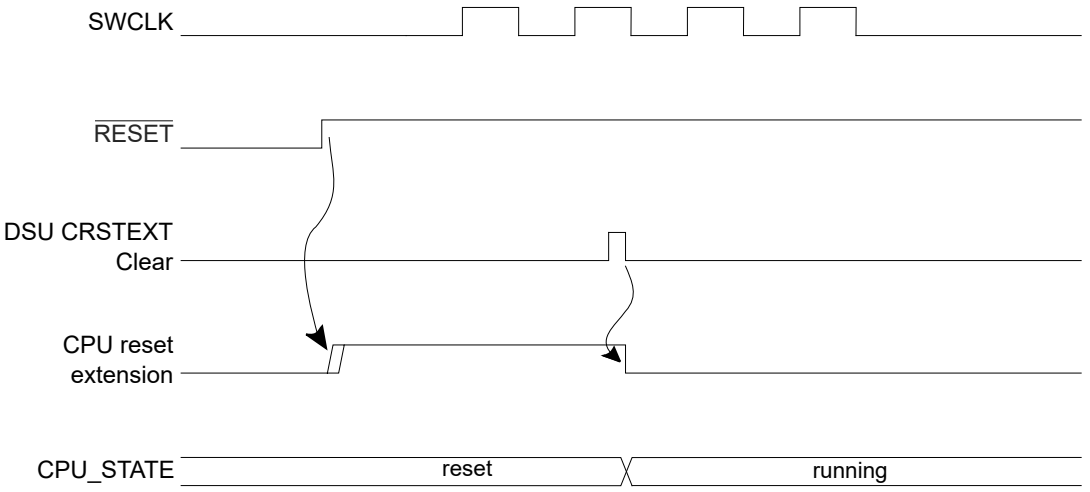
For more details on the ARM debug components, refer to the ARM Debug Interface v5 Architecture Specification.

### 12.6.2. CPU Reset Extension

“CPU Reset extension” refers to the extension of the Reset phase of the CPU core after the external Reset is released. This ensures that the CPU is not executing code at start-up while a debugger is connects to the system. The debugger is detected on a $\overline{\text{RESET}}$ release event when SWCLK is low. At start-up, SWCLK is internally pulled up to avoid false detection of a debugger if the SWCLK pin is left unconnected. When the CPU is held in the Reset extension phase, the CPU Reset Extension bit of the Status A register (STATUSA.CRSTEXT) is set. To release the CPU, write a '1' to STATUSA.CRSTEXT. STATUSA.CRSTEXT will then be set to '0'. Writing a '0' to STATUSA.CRSTEXT has no effect. For security reasons, it is not possible to release the CPU Reset extension when the device is protected by the NVMCTRL security bit (See NVMCTRL.CTRLB.CMD's Set Security Bit (SSB) command.). Trying to do so sets the Protection Error bit (PERR) of the Status A register (STATUSA.PERR).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 12-2.** Typical CPU Reset Extension Set and Clear Timing Diagram




**Related Links**

NVMCTRL – Nonvolatile Memory Controller

### 12.6.3. Debugger Probe Detection

#### 12.6.3.1. Cold Plugging

Cold-Plugging is the detection of a debugger when the system is in Reset. Cold-Plugging is detected when the CPU Reset extension is requested, as described above.

#### 12.6.3.2. Hot Plugging

Hot-Plugging is the detection of a debugger probe when the system is not in Reset. Hot-Plugging is not possible under Reset because the detector is reset when POR or $\overline{\text{RESET}}$ are asserted. Hot-Plugging is active when a SWCLK falling edge is detected. The SWCLK pad is multiplexed with other functions and the user must ensure that its default function is assigned to the debug system. If the SWCLK function is changed, the Hot-Plugging feature is disabled until a power reset or external Reset occurs. Availability of the Hot-Plugging feature can be read from the Hot-Plugging Enable bit of the Status B register (STATUSB.HPE).

**Figure 12-3.** Hot-Plugging Detection Timing Diagram




The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected. Once detected, the Debugger Present bit of the Status B register (STATUSB.DBGPRES) is set. For security reasons, Hot-Plugging is not available when the device is protected by the NVMCTRL security bit.

This detection requires that pads are correctly powered. Thus, at cold start-up, this detection cannot be done until POR is released. If the device is protected, Cold-Plugging is the only way to detect a debugger probe, and so the external Reset timing must be longer than the POR timing. If external


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Reset is deasserted before POR release, the user must retry the procedure above until it gets connected to the device.

**Related Links**

NVMCTRL – Nonvolatile Memory Controller

## 12.7. Chip Erase

Chip erase consists of removing all sensitive information stored in the chip and clearing the NVMCTRL security bit. Therefore, all volatile memories and the Flash memory (including the SmartEEPROM emulation area) will be erased. The Flash auxiliary rows, including the user row, will not be erased.

When the device is protected, the debugger must first reset the device in order to be detected. This ensures that internal registers are reset after the Protected state is removed. The chip erase operation is triggered by writing a '1' to the chip erase bit in the Control register (CTRL.CE). This command will be discarded if the DSU is protected by the Peripheral Access Controller (PAC) or when the chip erase command is locked (STATUSB.CELCK = 1). Once issued, the module clears volatile memories prior to erasing the Flash array. To ensure that the chip erase operation is completed, check the Done bit of the Status A register (STATUSA.DONE).

The chip erase operation depends on clocks and power management features that can be altered by the CPU. For that reason, it is recommended to issue a chip erase after a Cold-Plugging procedure to ensure that the device is in a known and Safe state.

The recommended sequence is as follows:

1. Issue the Cold-Plugging procedure (refer to Cold Plugging). The device then:
   a. Detects the debugger probe.
   b. Holds the CPU in Reset.
2. Issue the chip erase command by writing a '1' to CTRL.CE. The device then:
   a. Clears the system volatile memories.
   b. Erases the whole Flash array (including the SmartEEPROM emulation area, not including auxiliary rows).
   c. Erases the lock row, removing the NVMCTRL security bit protection.
3. Check for completion by polling STATUSA.DONE (read as '1' when completed).
4. Reset the device to let the NVMCTRL update the fuses.

## 12.8. Programming

Programming the Flash or RAM memories is only possible when the device is not protected by the NVMCTRL security bit. The programming procedure is as follows:

1. At power-up, $\overline{\text{RESET}}$ is driven low by a debugger. The on-chip regulator holds the system in a POR state until the input supply is above the POR threshold (refer to Power-on Reset (POR) characteristics). The system continues to be held in this Static state until the internally regulated supplies have reached a safe Operating state.
2. The PM starts, clocks are switched to the slow clock (Core Clock, System Clock, Flash Clock and any Bus Clocks that do not have clock gate control). Internal Resets are maintained due to the external Reset.
3. The debugger maintains a low level on SWCLK. $\overline{\text{RESET}}$ is released, resulting in a debugger Cold-Plugging procedure.
4. The debugger generates a clock signal on the SWCLK pin, the Debug Access Port (DAP) receives a clock.
5. The CPU remains in Reset due to the Cold-Plugging procedure; meanwhile, the rest of the system is released.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

6. A chip erase is issued to ensure that the Flash is fully erased prior to programming.
7. Programming is available through the AHB-AP.
8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$ or toggling power. Make sure that the SWCLK pin is high when releasing $\overline{\text{RESET}}$ to prevent extending the CPU Reset.

**Related Links**
NVMCTRL – Nonvolatile Memory Controller

## 12.9. Intellectual Property Protection

Intellectual property protection consists of restricting access to internal memories from external tools when the device is protected, and this is accomplished by setting the NVMCTRL security bit. This Protected state can be removed by issuing a chip erase (refer to Chip Erase). When the device is protected, read/write accesses using the AHB-AP are limited to the DSU address range and DSU commands are restricted. When issuing a chip erase, sensitive information is erased from volatile memory and Flash.

The DSU implements a security filter that monitors the AHB transactions inside the DAP. If the device is protected, then AHB-AP read/write accesses outside the DSU external address range are discarded, causing an error response that sets the ARM AHB-AP sticky error bits (refer to the ARM Debug Interface v5 Architecture Specification on www.arm.com).

The DSU is intended to be accessed either:

- Internally from the CPU, without any limitation, even when the device is protected
- Externally from a debug adapter, with some restrictions when the device is protected

For security reasons, DSU features have limitations when used from a debug adapter. To differentiate external accesses from internal ones, the first 0x100 bytes of the DSU register map has been mirrored at offset 0x100:

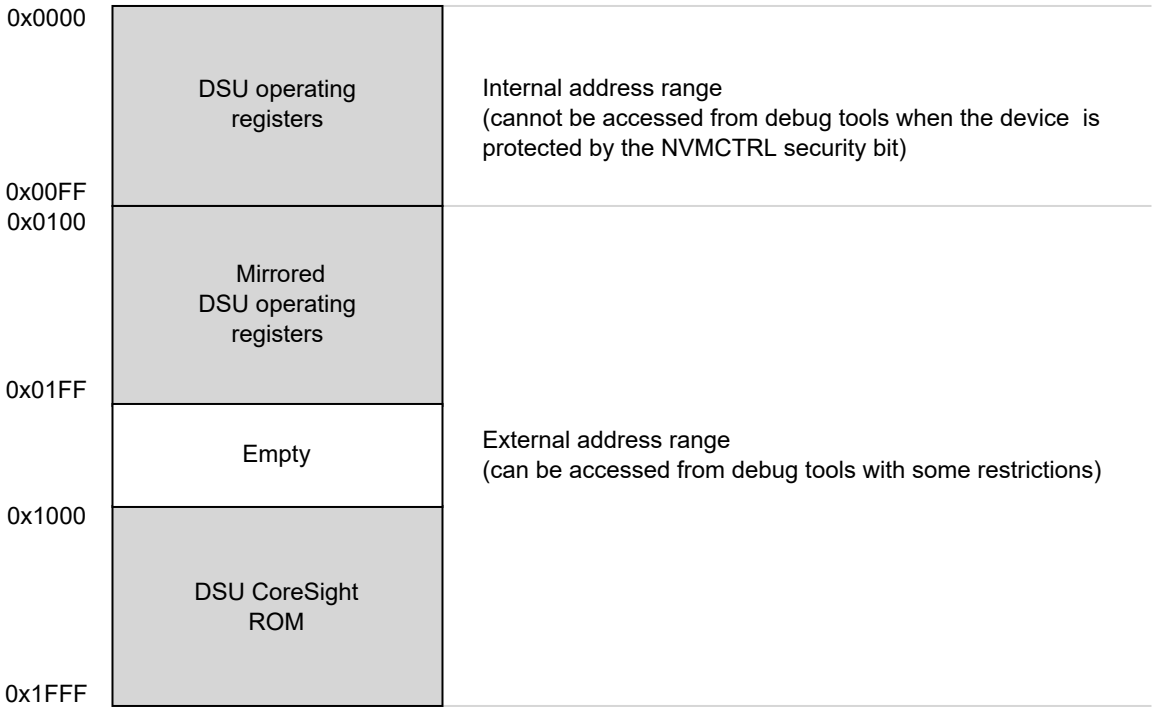
- The first 0x100 bytes form the internal address range
- The next 0x100 bytes form the external address range

When the device is protected, the DAP can only issue MEM-AP accesses in the DSU range 0x0100-0x2000.

The DSU Operating registers are located in the 0x0000-0x00FF area and remapped in 0x0100-0x01FF to differentiate accesses coming from a debugger and the CPU. If the device is protected and an access is issued in the region 0x0100-0x01FF, it is subject to security restrictions. For more information, refer to Table 12-1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 12-4.** APB Memory Mapping




Some features not activated by APB transactions are not available when the device is protected:

**Table 12-1.** Feature Availability Under Protection

| Features | Availability When the Device is Protected |
|---|---|
| CPU Reset Extension | Yes |
| Clear CPU Reset Extension | No |
| Debugger Cold-Plugging | Yes |
| Debugger Hot-Plugging | No |

### Related Links

NVMCTRL – Nonvolatile Memory Controller

## 12.10. Device Identification

Device identification relies on the ARM CoreSight component identification scheme, which allows the chip to be identified as a SAM device implementing a DSU. The DSU contains identification registers to differentiate the device.

### 12.10.1. CoreSight Identification

A system-level ARM® CoreSight™ ROM table is present in the device to identify the vendor and the chip identification method. Its address is provided in the MEM-AP BASE register inside the ARM Debug Access Port. The CoreSight ROM implements a 64-bit conceptual ID composed as follows from the PID0 to PID7 CoreSight ROM Table registers:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 12-5.** Conceptual 64-bit Peripheral ID

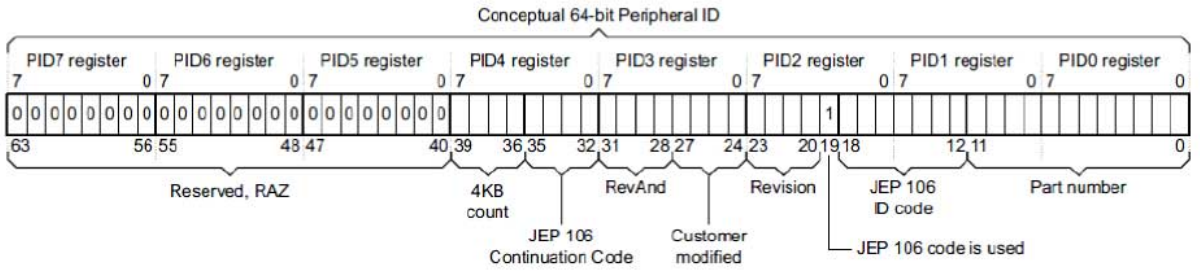



**Table 12-2.** Conceptual 64-Bit Peripheral ID Bit Descriptions

| Field | Size | Description | Location |
|---|---|---|---|
| JEP-106 CC code | 4 | Continuation code: 0x0 | PID4 |
| JEP-106 ID code | 7 | Device ID: 0x1F | PID1+PID2 |
| 4KB count | 4 | Indicates that the CoreSight component is a ROM: 0x0 | PID4 |
| RevAnd | 4 | Not used; read as 0 | PID3 |
| CUSMOD | 4 | Not used; read as 0 | PID3 |
| PARTNUM | 12 | Contains 0xCD0 to indicate that DSU is present | PID0+PID1 |
| REVISION | 4 | DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. If 0x0, this indicates that device identification can be completed by reading the Device Identification register (DID) | PID2 |

For more information, refer to the ARM Debug Interface Version 5 Architecture Specification.

### 12.10.2. Chip Identification Method

The DSU DID register identifies the device by implementing the following information:

- Processor identification
- Product family identification
- Product series identification
- Device select

## 12.11. Functional Description

### 12.11.1. Principle of Operation

The DSU provides memory services, such as CRC32 or MBIST that require almost the same interface. Hence, the Address, Length and Data registers (ADDR, LENGTH, DATA) are shared. These shared registers must be configured first; then a command can be issued by writing the Control register. When a command is ongoing, other commands are discarded until the current operation is completed. Hence, the user must wait for the STATUSA.DONE bit to be set prior to issuing another one.

### 12.11.2. Basic Operation

#### 12.11.2.1. Initialization

The module is enabled by enabling its clocks. For more details, refer to Clocks. The DSU registers can be PAC write-protected.

**Related Links**

PAC - Peripheral Access Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 12.11.2.2. Operation From a Debug Adapter

Debug adapters should access the DSU registers in the external address range 0x100 – 0x2000. If the device is protected by the NVMCTRL security bit, accessing the first 0x100 bytes causes the system to return an error. Refer to Intellectual Property Protection.

**Related Links**

NVMCTRL – Nonvolatile Memory Controller

#### 12.11.2.3. Operation From the CPU

There are no restrictions when accessing DSU registers from the CPU. However, the user should access DSU registers in the internal address range (0x0 – 0x100) to avoid external security restrictions. Refer to Intellectual Property Protection.

### 12.11.3. 32-bit Cyclic Redundancy Check CRC32

The DSU unit provides support for calculating a cyclic redundancy check (CRC32) value for a memory area (including Flash and AHB RAM).

When the CRC32 command is issued from:

- The internal range, the CRC32 can be operated at any memory location
- The external range, the CRC32 operation is restricted; DATA, ADDR, and LENGTH values are forced (see below)

**Table 12-3.** AMOD Bit Descriptions when Operating CRC32

| AMOD[1:0] | Short name | External range restrictions |
|---|---|---|
| 0 | ARRAY | CRC32 is restricted to the full Flash array area (SmartEEPROM emulation area not included) DATA forced to 0xFFFFFFFF before calculation (no seed) |
| 1 | EEPROM | CRC32 of the whole SmartEEPROM emulation area DATA forced to 0xFFFFFFFF before calculation (no seed) |
| 2-3 | Reserved | - |

The algorithm employed is the industry standard CRC32 algorithm using the generator polynomial 0xEDB88320 (reversed representation).

#### 12.11.3.1. Starting CRC32 Calculation

CRC32 calculation for a memory range is started after writing the start address into the Address register (ADDR) and the size of the memory range into the Length register (LENGTH). Both must be word-aligned.

The initial value used for the CRC32 calculation must be written to the Data register (DATA). This value will usually be 0xFFFFFFFF, but can be, for example, the result of a previous CRC32 calculation if generating a common CRC32 of separate memory blocks.

Once completed, the calculated CRC32 value can be read out of the Data register. The read value must be complemented to match standard CRC32 implementations or kept noninverted if used as starting point for subsequent CRC32 calculations.

The actual test is started by writing a '1' in the 32-bit Cyclic Redundancy Check bit of the Control register (CTRL.CRC). A running CRC32 operation can be canceled by resetting the module (writing '1' to CTRL.SWRST).

**Related Links**

NVMCTRL – Nonvolatile Memory Controller

#### 12.11.3.2. Interpreting the Results

The user should monitor the Status A register. When the operation is completed, STATUSA.DONE is set. Then the Bus Error bit of the Status A register (STATUSA.BERR) must be read to ensure that no bus error occurred.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.11.4. Debug Communication Channels

The Debug Communication Channels (DCCO and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger even if the device is protected by the NVMCTRL security bit. The registers can be used to exchange data between the CPU and the debugger, during run time as well as in Debug mode. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are accessible when the Protected state is active. When the device is protected, however, it is not possible to connect a debugger while the CPU is running (STATUSA.CRSTEXT is not writable and the CPU is held under Reset).

Two Debug Communication Channel status bits in the Status B registers (STATUS.DCCDx) indicate whether a new value has been written in DCC0 or DCC1. These bits, DCC0D and DCC1D, are located in the STATUSB registers. They are automatically set on write and cleared on read.

**Note:** The DCC0 and DCC1 registers are shared with the on-board memory testing logic (MBIST). Accordingly, DCC0 and DCC1 must not be used while performing MBIST operations.

**Related Links**
NVMCTRL – Nonvolatile Memory Controller

### 12.11.5. Debug Communication Channels DMA connection

The DCC0 and DCC1 registers can be used as a source or a destination of a DMA channel. The DSU generates one DMA request per Debug Communication Channels. The level of this DMA request is selectable writing the CFG.DCCDMALEVELx bit. Writing a 0 to this bit will configure the DMA request to trig on DCCx register empty. Writing a 1 to this bit will configure the DMA request to trig on DCCx register full.

### 12.11.6. Testing of On-Board Memories MBIST

The DSU implements a feature for automatic testing of memory, also known as MBIST (memory built-in self test). This is primarily intended for production test of on-board memories. MBIST cannot be operated from the external address range when the device is protected by the NVMCTRL security bit. If an MBIST command is issued when the device is protected, a protection error is reported in the Protection Error bit in the Status A register (STATUSA.PERR).

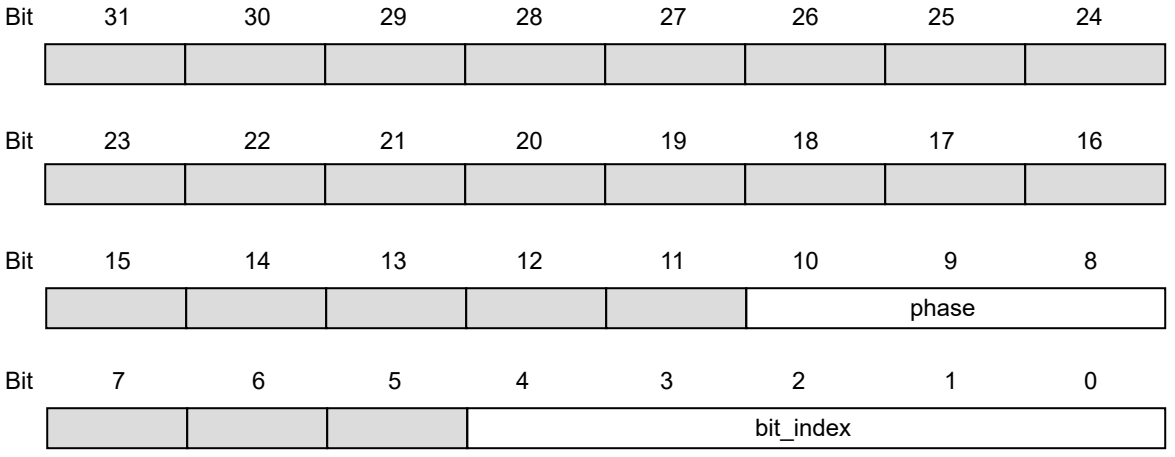
1. Algorithm
   The algorithm used for testing is a type of March algorithm called "March LR". This algorithm is able to detect a wide range of memory defects, while still keeping a linear run time. The algorithm is:
   a. Write entire memory to '0', in any order.
   b. Bit by bit read '0', write '1', in descending order.
   c. Bit by bit read '1', write '0', read '0', write '1', in ascending order.
   d. Bit by bit read '1', write '0', in ascending order.
   e. Bit by bit read '0', write '1', read '1', write '0', in ascending order.
   f. Read '0' from entire memory, in ascending order.

   The specific implementation used as a run time which depends on the CPU clock frequency and the number of bytes tested in the RAM. The detected faults are:
   - Address decoder faults
   - Stuck-at faults
   - Transition faults
   - Coupling faults
   - Linked Coupling faults


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

2. Starting MBIST
   To test a memory, you need to write the start address of the memory to the ADDR.ADDR bit field, and the size of the memory into the Length register.
   For best test coverage, an entire physical memory block should be tested at once. It is possible to test only a subset of a memory, but the test coverage will then be somewhat lower.
   The actual test is started by writing a '1' to CTRL.MBIST. A running MBIST operation can be canceled by writing a '1' to CTRL.SWRST.
3. Interpreting the Results
   The tester should monitor the STATUSA register. When the operation is completed, STATUSA.DONE is set. There are two different modes:
   - ADDR.AMOD=0: exit-on-error (default)
     In this mode, the algorithm terminates either when a fault is detected or on successful completion. In both cases, STATUSA.DONE is set. If an error was detected, STATUSA.FAIL will be set. User then can read the DATA and ADDR registers to locate the fault.
   - ADDR.AMOD=1: pause-on-error
     In this mode, the MBIST algorithm is paused when an error is detected. In such a situation, only STATUSA.FAIL is asserted. The state machine waits for user to clear STATUSA.FAIL by writing a '1' in STATUSA.FAIL to resume. Prior to resuming, user can read the DATA and ADDR registers to locate the fault.
4. Locating Faults
   If the test stops with STATUSA.FAIL set, one or more bits failed the test. The test stops at the first detected error. The position of the failing bit can be found by reading the following registers:
   - ADDR: Address of the word containing the failing bit
   - DATA: contains data to identify which bit failed, and during which phase of the test it failed. The DATA register will in this case contains the following bit groups:

**Figure 12-6.** DATA bits Description When MBIST Operation Returns an Error




- bit_index: contains the bit number of the failing bit
- phase: indicates which phase of the test failed and the cause of the error, as listed in the following table.

**Table 12-4.** MBIST Operation Phases

| Phase | Test actions |
|---|---|
| 0 | Write all bits to zero. This phase cannot fail. |
| 1 | Read '0', write '1', increment address |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 12-4.** MBIST Operation Phases (continued)

| Phase | Test actions |
|---|---|
| 2 | Read '1', write '0' |
| 3 | Read '0', write '1', decrement address |
| 4 | Read '1', write '0', decrement address |
| 5 | Read '0', write '1' |
| 6 | Read '1', write '0', decrement address |
| 7 | Read all zeros. bit_index is not used |

**Table 12-5.** AMOD Bit Descriptions for MBIST

| AMOD[1:0] | Description |
|---|---|
| 0x0 | Exit on Error |
| 0x1 | Pause on Error |
| 0x2, 0x3 | Reserved |

### Related Links

NVMCTRL – Nonvolatile Memory Controller
Product Memory Mapping Overview

## 12.11.7. System Services Availability when Accessed Externally and Device is Protected

External access: Access performed in the DSU address offset 0x200-0x1FFF range.

Internal access: Access performed in the DSU address offset 0x000-0x100 range.

**Table 12-6.** Available Features when Operated From The External Address Range and Device is Protected

| Features | Availability From The External Address Range and Device is Protected |
|---|---|
| Chip erase command and status | Yes |
| CRC32 | Yes, only full array or full SmartEEPROM |
| CoreSight Compliant Device identification | Yes |
| Debug communication channels | Yes |
| Testing of onboard memories (MBIST) | No |
| STATUSA.CRSTEXT clearing | No (STATUSA.PERR is set when attempting to do so) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 12.12. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRL | 7:0 | | | | CE | MBIST | CRC | | SWRST |
| 0x01 | STATUSA | 7:0 | | | | PERR | FAIL | BERR | CRSTEXT | DONE |
| 0x02 | STATUSB | 7:0 | | | CELCK | HPE | DCCD1 | DCCD0 | DBGPRES | PROT |
| 0x03 | Reserved | | | | | | | | | |
| 0x04 | ADDR | 7:0 | ADDR[5:0] | | | | | | AMOD[1:0] | |
| | | 15:8 | ADDR[13:6] | | | | | | | |
| | | 23:16 | ADDR[21:14] | | | | | | | |
| | | 31:24 | ADDR[29:22] | | | | | | | |
| 0x08 | LENGTH | 7:0 | LENGTH[5:0] | | | | | | | |
| | | 15:8 | LENGTH[13:6] | | | | | | | |
| | | 23:16 | LENGTH[21:14] | | | | | | | |
| | | 31:24 | LENGTH[29:22] | | | | | | | |
| 0x0C | DATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x10 | DCC0 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x14 | DCC1 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x18 | DID | 7:0 | DEVSEL[7:0] | | | | | | | |
| | | 15:8 | DIE[3:0] | | | | REVISION[3:0] | | | |
| | | 23:16 | FAMILY[0] | | SERIES[5:0] | | | | | |
| | | 31:24 | PROCESSOR[3:0] | | | | FAMILY[4:1] | | | |
| 0x1C | CFG | 7:0 | | | | ETBRAMEN | DCCDMALEVEL[1:0] | | LQOS[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 ... 0xEF | Reserved | | | | | | | | | |
| 0xF0 | DCFG0 | 7:0 | DCFG[7:0] | | | | | | | |
| | | 15:8 | DCFG[15:8] | | | | | | | |
| | | 23:16 | DCFG[23:16] | | | | | | | |
| | | 31:24 | DCFG[31:24] | | | | | | | |
| 0xF4 | DCFG1 | 7:0 | DCFG[7:0] | | | | | | | |
| | | 15:8 | DCFG[15:8] | | | | | | | |
| | | 23:16 | DCFG[23:16] | | | | | | | |
| | | 31:24 | DCFG[31:24] | | | | | | | |
| 0xF8 ... 0x0FFF | Reserved | | | | | | | | | |
| 0x1000 | ENTRY0 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x1004 | ENTRY1 | 7:0 | | | | | | | Reserved | Reserved |
| | | 15:8 | Reserved[3:0] | | | | | | | |
| | | 23:16 | Reserved[11:4] | | | | | | | |
| | | 31:24 | Reserved[19:12] | | | | | | | |
| 0x1008 | END | 7:0 | END[7:0] | | | | | | | |
| | | 15:8 | END[15:8] | | | | | | | |
| | | 23:16 | END[23:16] | | | | | | | |
| | | 31:24 | END[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x100C ... 0x1FCB | Reserved | | | | | | | | | |
| 0x1FCC | MEMTYPE | 7:0 | | | | | | | | SMEMP |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FD0 | PID4 | 7:0 | FKBC[3:0] | | | | JEPCC[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FD4 | PID5 | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FD8 | PID6 | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FDC | PID7 | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FE0 | PID0 | 7:0 | PARTNBL[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FE4 | PID1 | 7:0 | JEPIDCL[3:0] | | | | PARTNBH[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FE8 | PID2 | 7:0 | REVISION[3:0] | | | | JEPU | JEPIDCH[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FEC | PID3 | 7:0 | REVAND[3:0] | | | | CUSMOD[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FF0 | CID0 | 7:0 | PREAMBLEB0[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FF4 | CID1 | 7:0 | CCLASS[3:0] | | | | PREAMBLE[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FF8 | CID2 | 7:0 | PREAMBLEB2[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FFC | CID3 | 7:0 | PREAMBLEB3[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 12.13. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 12.13.1. Control

**Name:** CTRL
**Offset:** 0x0000
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | CE | MBIST | CRC | | SWRST |
| Access | | | | W | W | W | | W |
| Reset | | | | 0 | 0 | 0 | | 0 |

**Bit 4 – CE** Chip-Erase
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the Chip-Erase operation.

**Bit 3 – MBIST** Memory Built-In Self-Test
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the memory BIST algorithm.

**Bit 2 – CRC** 32-bit Cyclic Redundancy Check
Writing a '0' to this bit has no effect.
Writing a '1' to this bit starts the cyclic redundancy check algorithm.

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the module.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 12.13.2. Status A

**Name:** STATUSA
**Offset:** 0x0001
**Reset:** 0x00
**Property:** PAC Write Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | PERR | FAIL | BERR | CRSTEXT | DONE |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 4 – PERR** Protection Error
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Protection Error bit.
This bit is set when a command that is not allowed in Protected state is issued.

**Bit 3 – FAIL** Failure
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Failure bit.
This bit is set when a DSU operation failure is detected.

**Bit 2 – BERR** Bus Error
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Bus Error bit.
This bit is set when a bus error is detected.

**Bit 1 – CRSTEXT** CPU Reset Phase Extension
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CPU Reset Phase Extension bit.
This bit is set when a debug adapter Cold-Plugging is detected, which extends the CPU Reset phase.

**Bit 0 – DONE** Done
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Done bit.
This bit is set when a DSU operation is completed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.3. Status B

**Name:** STATUSB
**Offset:** 0x0002
**Reset:** 0x0x
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | CELCK | HPE | DCCD1 | DCCD0 | DBGPRES | PROT |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | x | x |

**Bit 5 – CELCK** Chip Erase Locked
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when Chip Erase is locked.
This bit is cleared when Chip Erase is unlocked.

**Bit 4 – HPE** Hot-Plugging Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when Hot-Plugging is enabled.
This bit is cleared when Hot-Plugging is disabled. This is the case when the SWCLK function is changed. Only a power-reset or a external reset can set it again.

**Bits 2, 3 – DCCD** Debug Communication Channel x Dirty
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when DCC is written.
This bit is cleared when DCC is read.

**Bit 1 – DBGPRES** Debugger Present
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set when a debugger probe is detected.
This bit is never cleared.

**Bit 0 – PROT** Protected
Writing a '0' to this bit has no effect.
Writing a '1' to this bit has no effect.
This bit is set at power-up when the device is protected (meaning the Security Bit is set).
This bit is reset at power-up when the device is not protected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.4. Address

**Name:** ADDR
**Offset:** 0x0004
**Reset:** 0x00000000
**Property:** PAC Write Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[29:22] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[21:14] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[13:6] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[5:0] | | | | | | AMOD[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:2 – ADDR[29:0]** Address
Initial word start address needed for memory operations.

**Bits 1:0 – AMOD[1:0]** Access Mode
The functionality of these bits is dependent on the operation mode.
Bit description when operating CRC32: refer to 32-bit Cyclic Redundancy Check CRC32
Bit description when testing onboard memories (MBIST): refer to Testing of On-Board Memories MBIST


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.5. Length

**Name:** LENGTH
**Offset:** 0x0008
**Reset:** 0x00000000
**Property:** PAC Write Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | LENGTH[29:22] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | LENGTH[21:14] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | LENGTH[13:6] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LENGTH[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bits 31:2 – LENGTH[29:0]** Length
Length in words needed for memory operations.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.6. Data

**Name:** DATA
**Offset:** 0x000C
**Reset:** 0x00000000
**Property:** PAC Write Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Data
Memory operation initial value or result value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.7. Debug Communication Channel x

**Name:** DCC
**Offset:** 0x10 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Data
Data register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.8. Device Identification

**Name:** DID
**Offset:** 0x0018
**Property:** -

The information in this register is related to the Ordering Information.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PROCESSOR[3:0] | | | | FAMILY[4:1] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | p | p | p | p | f | f | f | f |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FAMILY[0] | | SERIES[5:0] | | | | | |
| Access | R | | R | R | R | R | R | R |
| Reset | f | | s | s | s | s | s | s |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DIE[3:0] | | | | REVISION[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | d | d | d | d | r | r | r | r |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DEVSEL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

**Bits 31:28 – PROCESSOR[3:0]** Processor
The value of this field defines the processor used on the device.
**Note:** For this device, the value of this field is 0x6, corresponding to the Arm Cortex-M4F processor.

**Bits 27:23 – FAMILY[4:0]** Product Family
The value of this field corresponds to the product family part of the ordering code.
**Note:** For this device, the value of this field is 0x0 for SAM D5x and 0x3 for SAM E5x Family of base line microcontrollers.

**Bits 21:16 – SERIES[5:0]** Product Series
The value of this field corresponds to the product series part of the ordering code.

| Value | Description |
|---|---|
| 0x00 | Reserved |
| 0x01 | SAM E51 |
| 0x02 | Reserved |
| 0x03 | SAM E53 |
| 0x04 | SAM E54 |
| 0x05 | Reserved |
| 0x06 | SAM D51 |
| 0x07 - 0x3F | Reserved |

**Bits 15:12 – DIE[3:0]** Die Number
Identifies the die family.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 11:8 – REVISION[3:0]** Revision Number

Identifies the die revision number. Refer the product family silicon errata and data sheet clarification document for further information.

**Note:** The device variant (last letter of the ordering number) is independent of the die revision (DSU.DID.REVISION): The device variant denotes functional differences, whereas the die revision marks evolution of the die.

**Bits 7:0 – DEVSEL[7:0]** Device Selection

This bit field identifies a device within a product family and product series.

**Table 12-7.** SAM D51 Device Selection

| Value | Description |
|---|---|
| 0x00 | 1 MB Flash / 256 KB SRAM / 120-pin or 128-pin |
| 0x01 | 512 KB Flash / 192 KB SRAM / 120-pin or 128-pin |
| 0x02 | 1 MB Flash / 256 KB SRAM / 100-pin |
| 0x03 | 512 KB Flash / 192 KB SRAM / 100-pin |
| 0x04 | 1 MB Flash / 256 KB SRAM / 64-pin |
| 0x05 | 512 KB Flash / 192 KB SRAM / 64-pin |
| 0x06 | 256 KB Flash / 128 KB SRAM / 64-pin |
| 0x07 | 512 KB Flash / 192 KB SRAM / 48-pin |
| 0x08 | 256 KB Flash / 128 KB SRAM / 48-pin |
| 0x09 - 0xFF | Reserved |

**Table 12-8.** SAM E51 Device Selection

| Value | Description |
|---|---|
| 0x00 | 1 MB Flash / 256 KB SRAM / 100-pin |
| 0x01 | 512 KB Flash / 192 KB SRAM / 100-pin |
| 0x02 | 512 KB Flash / 192 KB SRAM / 64-pin |
| 0x03 | 256 KB Flash / 128 KB SRAM / 64-pin |
| 0x04 | 1 MB Flash / 256 KB SRAM / 64-pin |
| 0x05 | Reserved |
| 0x06 | 256 KB Flash / 128 KB SRAM / 48-pin |
| 0x07 - 0xFF | Reserved |

**Table 12-9.** SAM E53 Device Selection

| Value | Description |
|---|---|
| 0x00 - 0x01 | Reserved |
| 0x02 | 1 MB Flash / 256 KB SRAM / 100-pin |
| 0x03 | 512 KB Flash / 192 KB SRAM / 100-pin |
| 0x04 | 1 MB Flash / 256 KB SRAM / 64-pin |
| 0x05 | 512 KB Flash / 192 KB SRAM / 64-pin |
| 0x06 | 256 KB Flash / 128 KB SRAM / 64-pin |
| 0x07 - 0xFF | Reserved |

**Table 12-10.** SAM E54 Device Selection

| Value | Description |
|---|---|
| 0x00 | 1 MB Flash / 256 KB SRAM / 120-pin or 128-pin |
| 0x01 | 512 KB Flash / 192 KB SRAM / 120-pin or 128-pin |
| 0x02 | 1 MB Flash / 256 KB SRAM / 100-pin |
| 0x03 | 512 KB Flash / 192 KB SRAM / 100-pin |
| 0x04 - 0xFF | Reserved |

### Related Links

Ordering Information


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.9. Configuration

**Name:** CFG
**Offset:** 0x1C
**Reset:** 0x00000002
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | ETBRAMEN | DCCDMALEVEL[1:0] | | LQOS[1:0] | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 1 | 0 |

**Bit 4 – ETBRAMEN** Trace Control
ETB Ram Enable Writing a one to this bit will reserve the first 32KB of the RAM for the Trace ETB ram buffer. Refer to *Memories / SRAM Memory Configuration* section for details.

**Bits 3:2 – DCCDMALEVEL[1:0]** DMA Trigger Level

| Value | Description |
|---|---|
| 0x0 | DMA Trigger rises when DCC is empty. |
| 0x1 | DMA Trigger rises when DCC is full. |
| 0x2 – 0x3 | Reserved |

**Bits 1:0 – LQOS[1:0]** Latency Quality Of Service
These bits define the priority access during the memory access. Refer to *SRAM Quality of Service*.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.10. Device Configuration

**Name:** DCFG
**Offset:** 0xF0 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DCFG[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DCFG[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DCFG[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DCFG[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DCFG[31:0]** Device Configuration


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.11. CoreSight ROM Table Entry x

**Name:** ENTRY0
**Offset:** 0x1000
**Reset:** 0x9F0FC002
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | x | x | x | x | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | FMT | EPRES |
| Access | | | | | | | R | R |
| Reset | | | | | | | 1 | x |

**Bits 31:12 – ADDOFF[19:0]** Address Offset
The base address of the component, relative to the base address of this ROM table.

**Bit 1 – FMT** Format
Always reads as '1', indicating a 32-bit ROM table.

**Bit 0 – EPRES** Entry Present
This bit indicates whether an entry is present at this location in the ROM table.
This bit is set at power-up if the device is not protected indicating that the entry is not present.
This bit is cleared at power-up if the device is not protected indicating that the entry is present.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.12. CoreSight ROM Table Entry 1

**Name:** ENTRY1
**Offset:** 0x1004
**Reset:** -
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | Reserved[19:12] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | Reserved[11:4] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | Reserved[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | x | x | x | x | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | Reserved | Reserved |
| Access | | | | | | | R | R |
| Reset | | | | | | | x | x |

**Bits 31:12 – Reserved[19:0]**

**Bit 1 – Reserved**

**Bit 0 – Reserved**


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.13. CoreSight ROM Table End

**Name:** END
**Offset:** 0x1008
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | END[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | END[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | END[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | END[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – END[31:0]** End Marker
Indicates the end of the CoreSight ROM table entries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.14. CoreSight ROM Table Memory Type

**Name:** MEMTYPE
**Offset:** 0x1FCC
**Reset:** 0x0000000x
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SMEMP |
| Access | | | | | | | | R |
| Reset | | | | | | | | x |

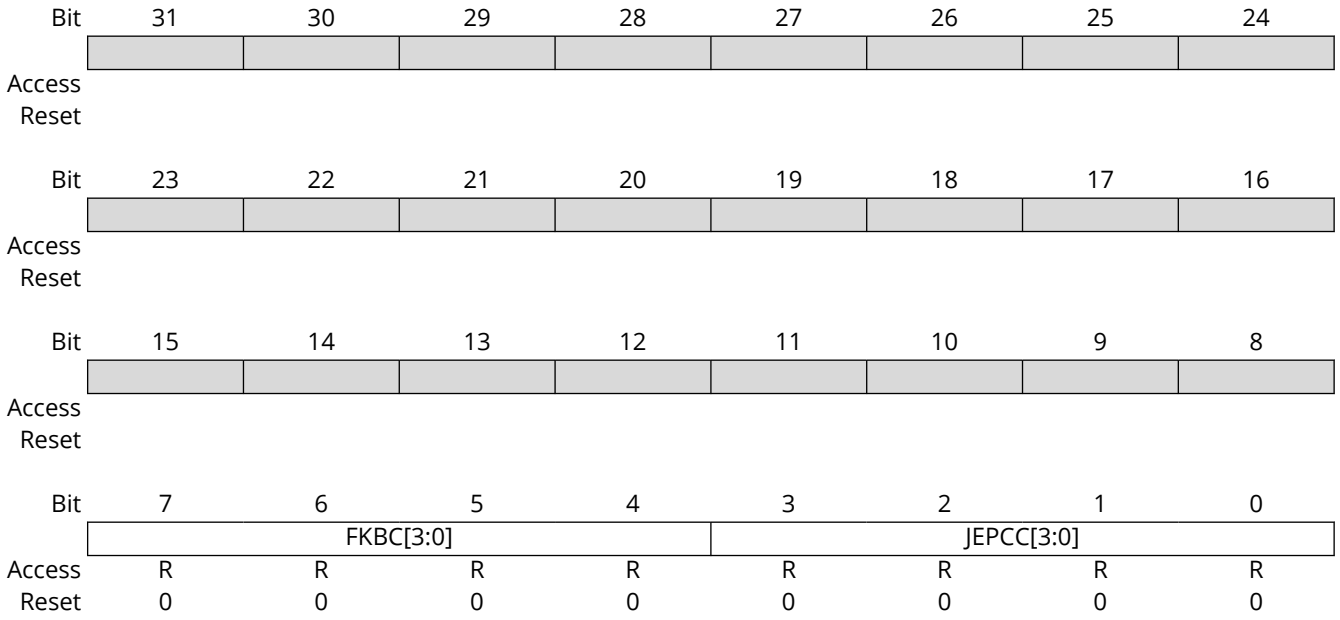
**Bit 0 – SMEMP** System Memory Present

This bit indicates whether system memory is present on the bus that connects to the ROM table.
This bit is set at power-up if the device is not protected, indicating that the system memory is accessible from a debug adapter.
This bit is cleared at power-up if the device is protected, indicating that the system memory is not accessible from a debug adapter.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.15. Peripheral Identification 4

**Name:** PID4
**Offset:** 0x1FD0
**Reset:** 0x00000000
**Property:** -



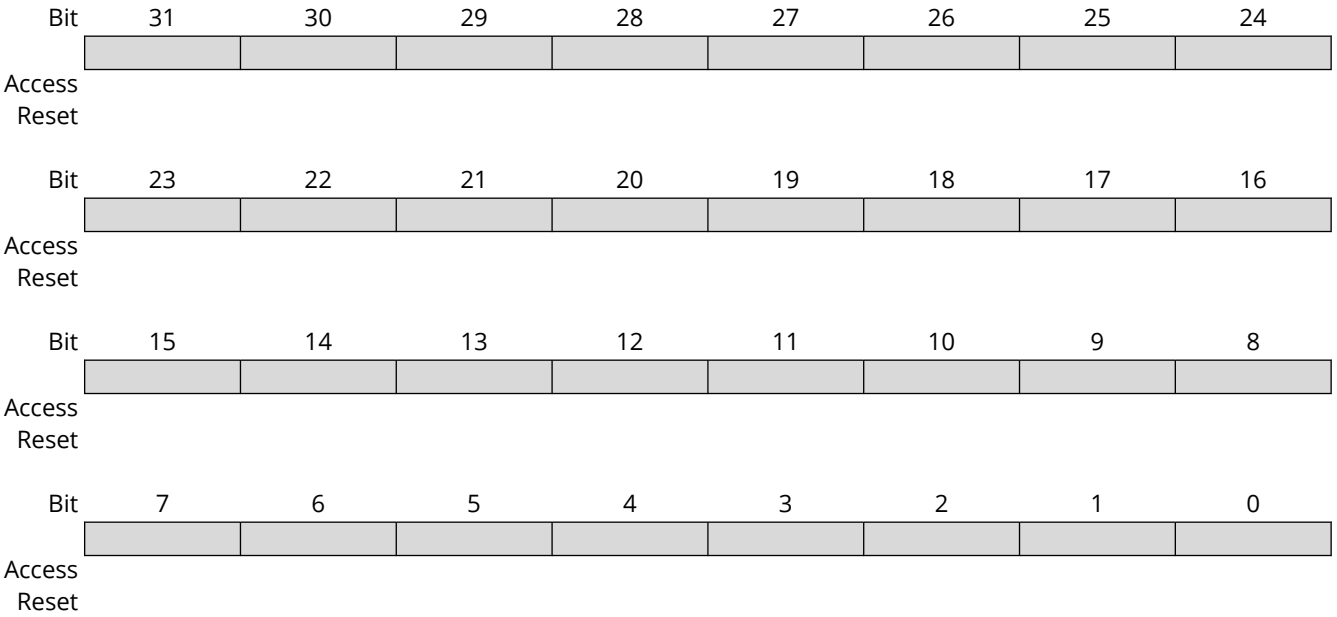

**Bits 7:4 – FKBC[3:0]** 4KB Count
These bits will always return zero when read, indicating that this debug component occupies one 4KB block.

**Bits 3:0 – JEPCC[3:0]** JEP-106 Continuation Code
These bits will always return zero when read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.16. Peripheral Identification 7

**Name:** PID7
**Offset:** 0x1FDC
**Reset:** 0x00000000
**Property:** Read-Only



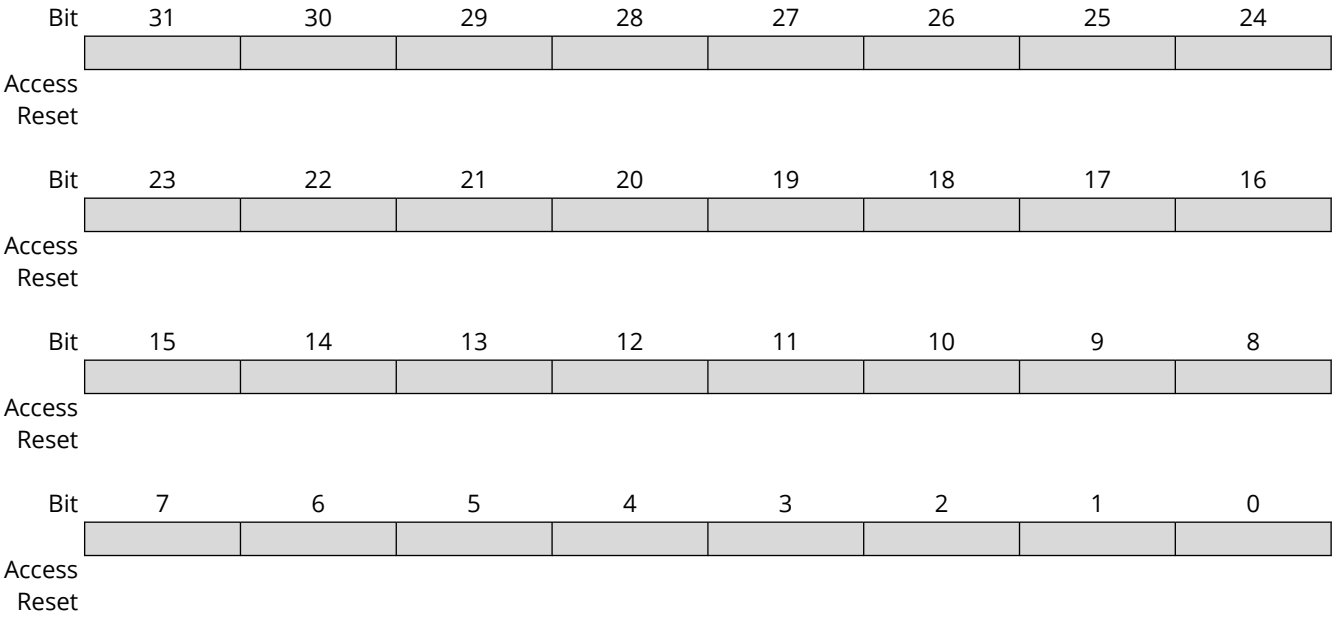


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 12.13.17. Peripheral Identification 6

**Name:** PID6
**Offset:** 0x1FD8
**Reset:** 0x00000000
**Property:** Read-Only



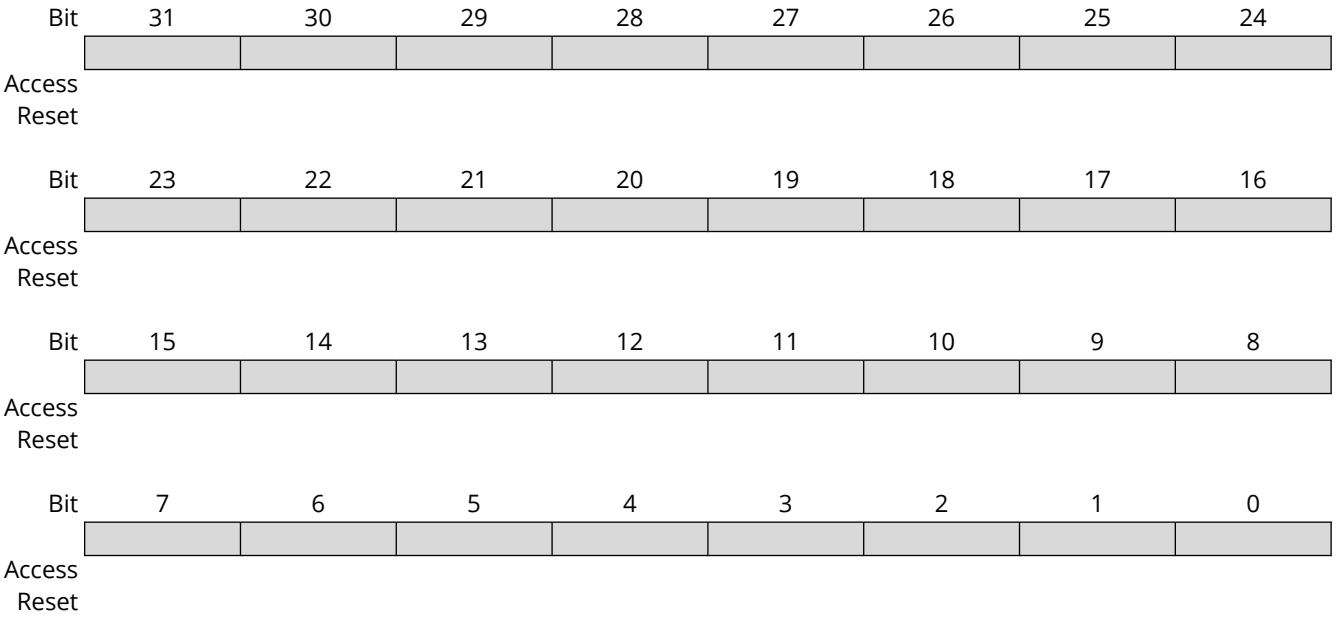


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 12.13.18. Peripheral Identification 5

**Name:** PID5
**Offset:** 0x1FD4
**Reset:** 0x00000000
**Property:** Read-Only



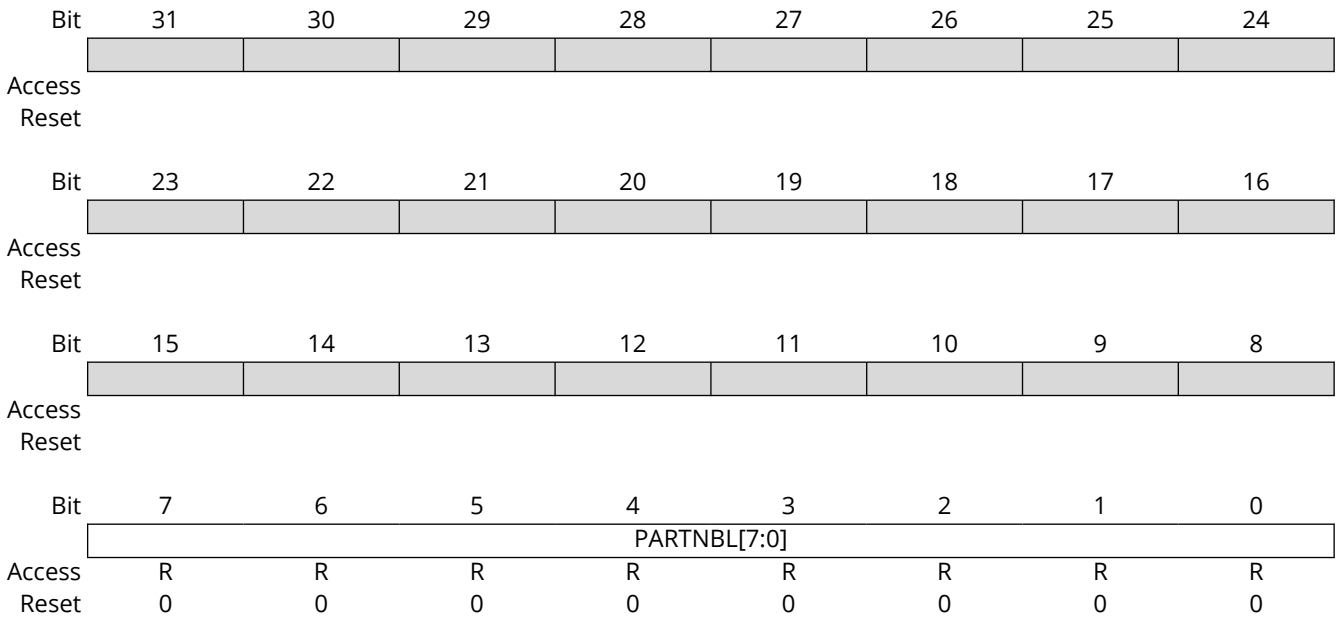


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 12.13.19. Peripheral Identification 0

**Name:** PID0
**Offset:** 0x1FE0
**Reset:** 0x00000000
**Property:** -



| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PARTNBL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – PARTNBL[7:0]** Part Number Low
These bits will always return 0xD0 when read, indicating that this device implements a DSU module instance.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.20. Peripheral Identification 1

**Name:** PID1
**Offset:** 0x1FE4
**Reset:** 0x000000FC
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | JEPIDCL[3:0] | | | | PARTNBH[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

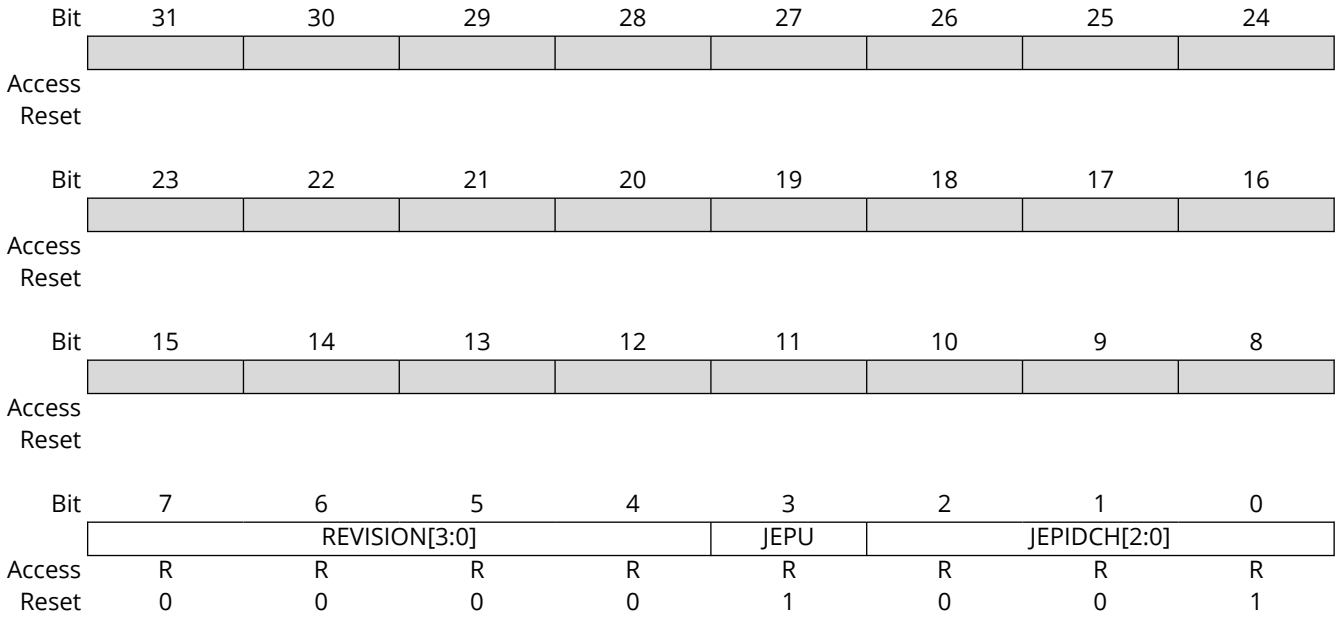
**Bits 7:4 – JEPIDCL[3:0]** Low Part of the JEP-106 Identity Code
These bits will always return 0xF when read (JEP-106 identity code is 0x1F).

**Bits 3:0 – PARTNBH[3:0]** Part Number High
These bits will always return 0xC when read, indicating that this device implements a DSU module instance.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.21. Peripheral Identification 2

**Name:** PID2
**Offset:** 0x1FE8
**Reset:** 0x00000009
**Property:** -




**Bits 7:4 – REVISION[3:0]** Revision Number
Revision of the peripheral. Starts at 0x0 and increments by one at both major and minor revisions.

**Bit 3 – JEPU** JEP-106 Identity Code is Used
This bit will always return one when read, indicating that JEP-106 code is used.

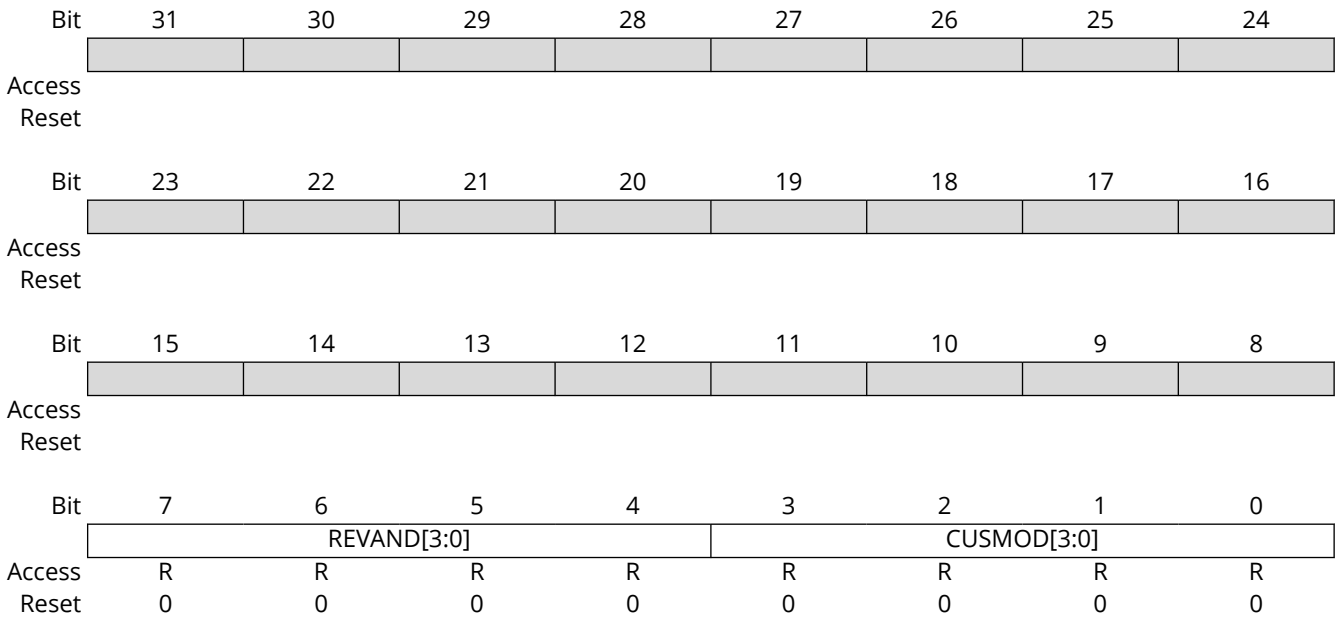
**Bits 2:0 – JEPIDCH[2:0]** JEP-106 Identity Code High
These bits will always return 0x1 when read, (JEP-106 identity code is 0x1F).


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 12.13.22. Peripheral Identification 3

**Name:** PID3
**Offset:** 0x1FEC
**Reset:** 0x00000000
**Property:** -



| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | REVAND[3:0] | | | | CUSMOD[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:4 – REVAND[3:0]** Revision Number
These bits will always return 0x0 when read.

**Bits 3:0 – CUSMOD[3:0]** ARM CUSMOD
These bits will always return 0x0 when read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.23. Component Identification 0

**Name:** CID0
**Offset:** 0x1FF0
**Reset:** 0x0000000D
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PREAMBLEB0[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

**Bits 7:0 – PREAMBLEB0[7:0]** Preamble Byte 0
These bits will always return 0x0000000D when read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.24. Component Identification 1

**Name:** CID1
**Offset:** 0x1FF4
**Reset:** 0x00000010
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCLASS[3:0] | | | | PREAMBLE[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

**Bits 7:4 – CCLASS[3:0]** Component Class
These bits will always return 0x1 when read indicating that this ARM CoreSight component is ROM table (refer to the ARM Debug Interface v5 Architecture Specification at http://www.arm.com).

**Bits 3:0 – PREAMBLE[3:0]** Preamble
These bits will always return 0x00 when read.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 12.13.25. Component Identification 2

**Name:** CID2
**Offset:** 0x1FF8
**Reset:** 0x00000005
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PREAMBLEB2[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

**Bits 7:0 – PREAMBLEB2[7:0]** Preamble Byte 2
These bits will always return 0x00000005 when read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 12.13.26. Component Identification 3

**Name:** CID3
**Offset:** 0x1FFC
**Reset:** 0x000000B1
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PREAMBLEB3[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

**Bits 7:0 – PREAMBLEB3[7:0]** Preamble Byte 3
These bits will always return 0x000000B1 when read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 13. Clock System

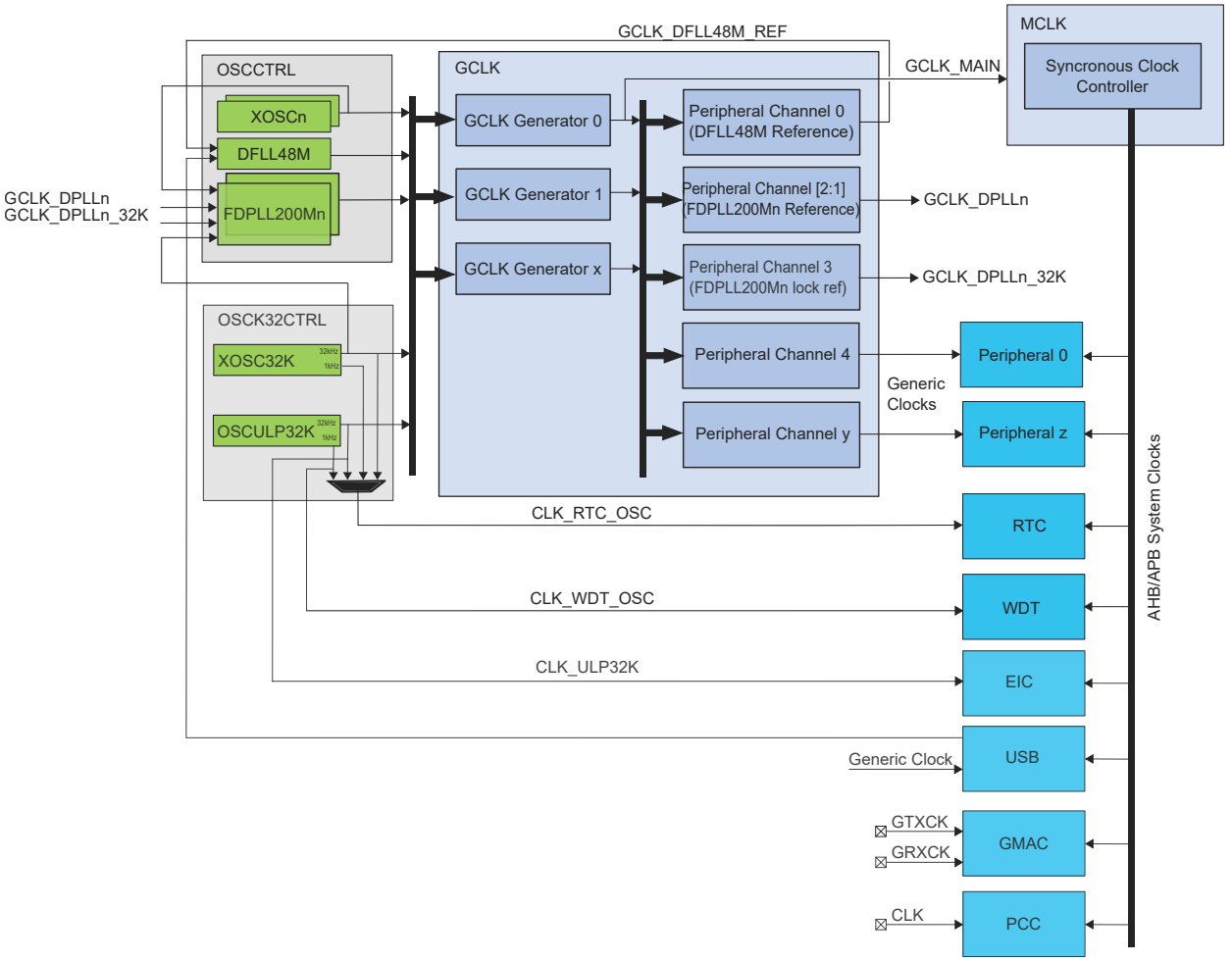
This chapter summarizes the clock distribution and terminology in the SAM D5x/E5x device. It does not explain every detail of its configuration. For in-depth documentation, see the respective peripherals descriptions and the *Generic Clock* documentation.

**Related Links**

GCLK - Generic Clock Controller
MCLK – Main Clock

## 13.1. Clock Distribution

**Figure 13-1.** Clock Distribution




The SAM D5x/E5x clock system consists of the following:

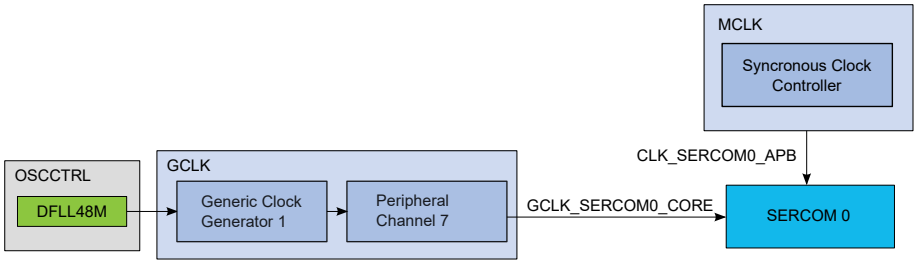
- Clock sources – Oscillators controlled by OSCCTRL and OSC32KCTRL.
  - A clock source provides a time base that is used by other components, such as Generic Clock Generators. Example clock sources include the external crystal oscillators (XOSC0 and XOSC1) and the Digital Frequency Locked Loop (DFLL48M).
- Generic Clock Controller (GCLK) – Which generates, controls and distributes asynchronous clocks consisting of the following:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - Generic Clock Generators: These are programmable prescalers that can use any of the system clock sources as a time base. The Generic Clock Generator 0 generates the clock signal GCLK_MAIN, which is used by the Power Manager and the Main Clock (MCLK) module, which in turn generates synchronous clocks.
  - Generic Clocks: These are clock signals generated by Generic Clock Generators and output by the Peripheral Channels, and serve as clocks for the peripherals of the system. Multiple instances of a peripheral will typically have a separate Generic Clock for each instance. Generic Clock 0 serves as the clock source for the DFLL48M clock input (when multiplying another clock source).
- Main Clock Controller (MCLK)
  - The MCLK generates and controls the synchronous clocks on the system. This includes the CPU, bus clocks (APB, AHB) as well as the synchronous (to the CPU) user interfaces of the peripherals. It contains clock masks that can turn on/off the user interface of a peripheral as well as prescalers for the CPU and bus clocks.

The figure below shows an example where SERCOM0 is clocked by the DFLL48M in open-loop mode. The DFLL48M is enabled, the Generic Clock Generator 1 uses the DFLL48M as its clock source and feeds into Peripheral Channel 7. The Generic Clock 7, also called GCLK_SERCOM0_CORE, which is connected to SERCOM0. The SERCOM0 interface, clocked by CLK_SERCOM0_APB, has been unmasked in the APBC Mask register in the MCLK.

**Figure 13-2.** Example of SERCOM Clock




To customize the clock distribution, refer to these registers and bit fields:

- The source oscillator for a Generic Clock Generator n is selected by writing to the Source bit field in the Generator Control n register (GCLK.GENCTRLn.SRC).
- A Peripheral Channel m can be configured to use a specific Generic Clock Generator by writing to the Generic Clock Generator bit field in the respective Peripheral Channel m register (GCLK.PCHCTRLm.GEN).
- The Peripheral Channel number, *m*, is fixed for a given peripheral. See Table 'PCHCTRLm Mapping' in the description of GCLK.PCHCTRLm.
- The AHB clocks are enabled and disabled by writing to the respective bit in the AHB Mask register (MCLK.AHBMASK).
- The APB clocks are enabled and disabled by writing to the respective bit in the APB x Mask registers (MCLK.APBxMASK).

**Related Links**

Clocks after Reset

## 13.2. Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, that is they are clocked from different clock sources and with different clock speeds, some peripheral accesses by the CPU need


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

to be synchronized. In this case the peripheral includes a Synchronization Busy (SYNCBUSY) register that can be used to check if a sync operation is in progress.

For a general description, refer to Register Synchronization. Some peripherals have specific properties described in their individual sub-chapter “Synchronization”.

In the data sheet, references to Synchronous Clocks are referring to the CPU and bus clocks (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).

**Related Links**
Synchronization

## 13.3. Register Synchronization

### 13.3.1. Overview

All peripherals are composed of one digital bus interface connected to the APB or AHB bus and running from a corresponding clock in the Main Clock domain, and one peripheral core running from the peripheral Generic Clock (GCLK).

Communication between these clock domains must be synchronized. This mechanism is implemented in hardware, so the synchronization process takes place even if the peripheral generic clock is running from the same clock source and on the same frequency as the bus interface.

All registers in the bus interface are accessible without synchronization.

All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

Each individual register description will have the properties ‘Read-Synchronized’ and/or ‘Write-Synchronized’ if a register is synchronized.

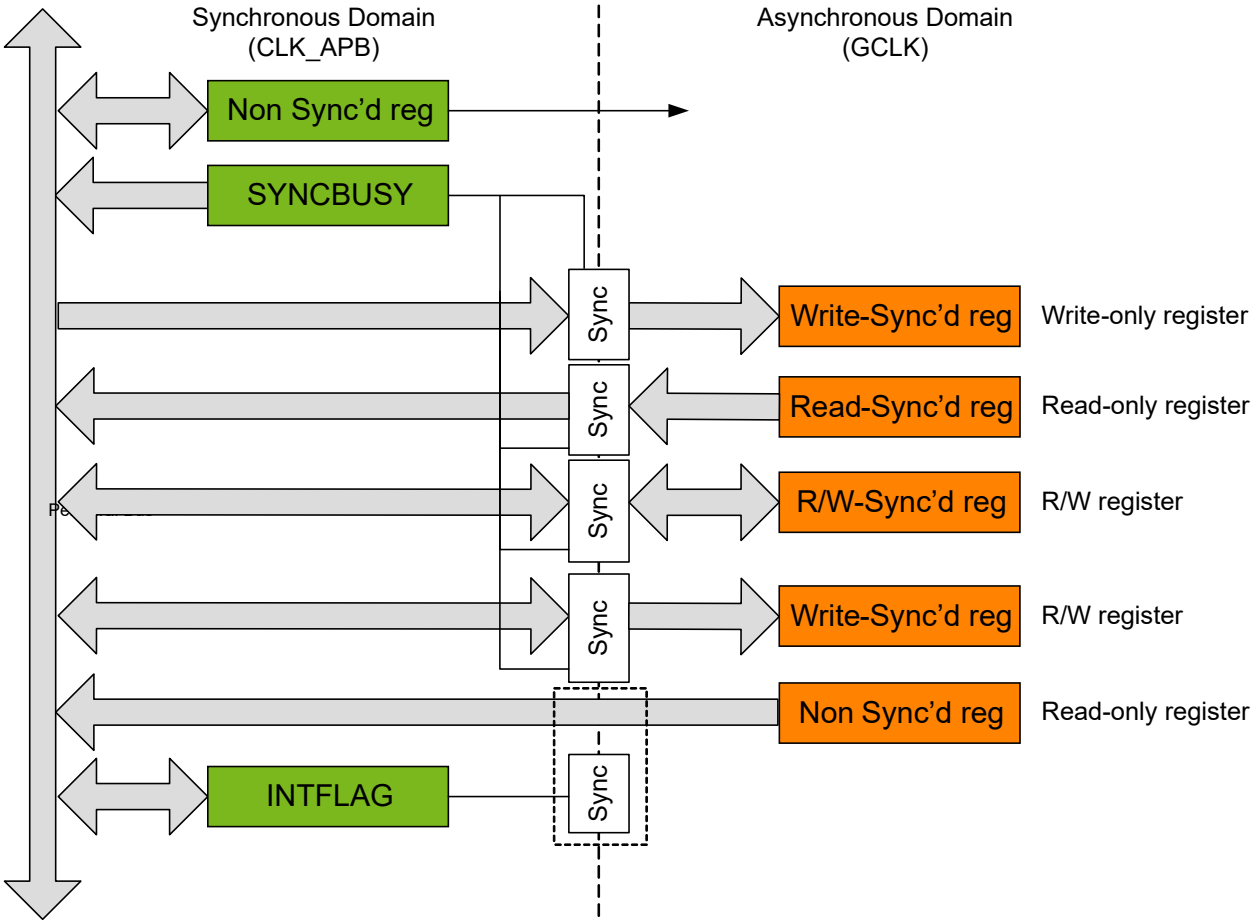
As shown in the figure below, each register that requires synchronization has its individual synchronizer and its individual synchronization status bit in the Synchronization Busy register (SYNCBUSY).

**Notes:**

1. For registers requiring both read-synchronization and write-synchronization, the corresponding bit in the SYNCBUSY registered is shared.
2. Due to asynchronicity between the main clock domain and the peripheral clock domains, all Read-Synchronized registers must ensure synchronization is completed before accessing them after waking up from Standby sleep mode. The respective bit in the SYNCBUSY register will be cleared when the write-synchronization is complete.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 13-3.** Register Synchronization Overview




### 13.3.2. General Write Synchronization

Write-Synchronization is triggered by writing to a register in the peripheral clock domain (GCLK). The respective bit in the Synchronization Busy register (SYNCBUSY) will be set when the write-synchronization starts and cleared when the write-synchronization is complete. Refer also to Synchronization Delay.

When write-synchronization is ongoing for a register, any subsequent write attempts to this register will be discarded, and an error will be reported though the Peripheral Access Controller (PAC).

**Example:**

REGA, REGB are 8-bit core registers. REGC is a 16-bit core register.

| Offset | Register |
|---|---|
| 0x00 | REGA |
| 0x01 | REGB |
| 0x02 | REGC |
| 0x03 | |

Synchronization is per register, so multiple registers can be synchronized in parallel. Consequently, after REGA (8-bit access) was written, REGB (8-bit access) can be written immediately without error.

REGC (16-bit access) can be written without affecting REGA or REGB. If REGC is written to in two consecutive 8-bit accesses without waiting for synchronization, the second write attempt will be discarded and an error is generated through the PAC.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

A 32-bit access to offset 0x00 will write all three registers. Note that REGA, REGB and REGC can be updated at different times because of independent write synchronization.

### 13.3.3. General Read Synchronization

Read-synchronized registers are synchronized each time the register value is updated but the corresponding SYNCBUSY bits are not set. Reading a read-synchronized register does not start a new synchronization, it returns the last synchronized value.
**Note:** The corresponding bits in SYNCBUSY will automatically be set when the device wakes up from sleep because read-synchronized registers need to be synchronized. Therefore reading a read-synchronized register before its corresponding SYNCBUSY bit is cleared will return the last synchronized value before sleep mode.
However, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be executed successfully. If concurrent read and write access is detected, the read is discarded and a new synchronization will start.

### 13.3.4. Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronisation Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, i.e. when all bits in SYNCBUSY are '0'.

### 13.3.5. Write Synchronization for CTRLA.ENABLE

Setting the Enable bit in a module's Control A register (CTRLA.ENABLE) will trigger write-synchronization and set SYNCBUSY.ENABLE.

CTRLA.ENABLE will read its new value immediately after being written.

SYNCBUSY.ENABLE will be cleared by hardware when the operation is complete.

The Synchronization Ready interrupt (if available) cannot be used to enable write-synchronization.

### 13.3.6. Write-Synchronization for Software Reset Bit

Setting the Software Reset bit in CTRLA (CTRLA.SWRST=1) will trigger write-synchronization and set SYNCBUSY.SWRST. When writing a ‘1’ to the CTRLA.SWRST bit it will immediately read as ‘1’.

CTRL.SWRST and SYNCBUSY.SWRST will be cleared by hardware when the peripheral has been reset.

Writing a '0' to the CTRL.SWRST bit has no effect.

The Ready interrupt (if available) cannot be used for Software Reset write-synchronization.

**Note:** Not all peripherals have the SWRST bit in the respective CTRLA register.

### 13.3.7. Synchronization Delay

The synchronization will delay write and read accesses by a certain amount. This delay *D* is within the range of:

$5\times P_{GCLK} + 2\times P_{APB} < D < 6\times P_{GCLK} + 3\times P_{APB}$

Where $P_{GCLK}$ is the period of the generic clock and $P_{APB}$ is the period of the peripheral bus clock. A normal peripheral bus register access duration is $2\times P_{APB}$.

## 13.4. Enabling a Peripheral

In order to enable a peripheral that is clocked by a Generic Clock, the following parts of the system needs to be configured:

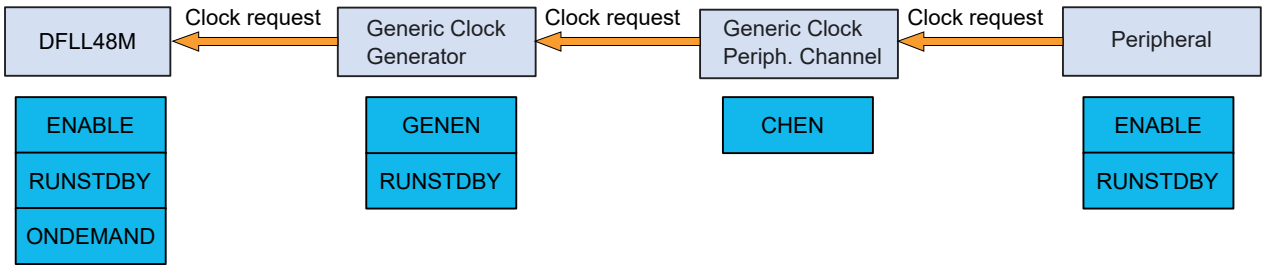
- A running Clock Source
- A clock from the Generic Clock Generator must be configured to use one of the running Clock Sources, and the Generator must be enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The Peripheral Channel that provides the Generic Clock signal to the peripheral must be configured to use a running Generic Clock Generator, and the Generic Clock must be enabled.
- The user interface of the peripheral needs to be unmasked in the PM. If this is not done the peripheral registers will read all 0's and any writing attempts to the peripheral will be discarded.

## 13.5. On Demand Clock Requests

**Figure 13-4.** Clock Request Routing




All clock sources in the system can be run in an on-demand mode: the clock source is in a stopped state unless a peripheral is requesting the clock source. Clock requests propagate from the peripheral, via the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started/kept running. As soon as the clock source is no longer needed and no peripheral has an active request, the clock source will be stopped until requested again.

The clock request can reach the clock source only if the peripheral, the generic clock and the clock from the Generic Clock Generator in-between are enabled. The time taken from a clock request being asserted to the clock source being ready is dependent on the clock source startup time, clock source frequency as well as the divider used in the Generic Clock Generator. The total startup time $T_{start}$ from a clock request until the clock is available for the peripheral is between:

$T_{start_max}$ = Clock source startup time + 2 × clock source periods + 2 × divided clock source periods

$T_{start_min}$ = Clock source startup time + 1 × clock source period + 1 × divided clock source period

The time between the last active clock request stopped and the clock is shut down, $T_{stop}$, is between:

$T_{stop_min}$ = 1 × divided clock source period + 1 × clock source period

$T_{stop_max}$ = 2 × divided clock source periods + 2 × clock source periods

The On-Demand function can be disabled individually for each clock source by clearing the ONDEMAND bit located in each clock source controller. Consequently, the clock will always run whatever the clock request status is. This has the effect of removing the clock source startup time at the cost of power consumption.

The clock request mechanism can be configured to work in standby mode by setting the RUNSDTBY bits of the modules, see Figure 13-4.

## 13.6. Power Consumption vs. Speed

When targeting for either a low-power or a fast acting system, some considerations have to be taken into account due to the nature of the asynchronous clocking of the peripherals:

If clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At the same time the synchronization to the synchronous (CPU) clock domain is dependent on the peripheral clock speed, and will take longer with a slower peripheral clock. This will cause worse response times and longer synchronization delays.

## 13.7. Clocks after Reset

On any Reset the synchronous clocks start to their initial state:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- DFLL48M is enabled and configured to run at 48MHz
- Generic Generator 0 uses DFLL48M as source and generates GCLK_MAIN
- CPU and BUS clocks are undivided

On a Power-on Reset, the 32KHz clock sources are reset and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except
  - Generator 0 is using DFLL48M at 48MHz as source and generates GCLK_MAIN
- All Peripheral Channels in GCLK are disabled.

On a User Reset the GCLK module starts to its initial state, except for:

- Generic Clocks that are write-locked, i.e., the according WRTLOCK is set to 1 prior to Reset

**Related Links**
RSTC – Reset Controller


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 14. GCLK - Generic Clock Controller

## 14.1. Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller (GCLK) features 12 Generic Clock Generators [11:0] that can provide a wide range of clock frequencies.

Generators can be set to use different external and internal oscillators as source. The clock of each Generator can be divided. The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules, as shown in Figure 14-2. The number of Peripheral Clocks depends on how many peripherals the device has.
**Note:** The Generator 0 is always the direct source of the GCLK_MAIN signal.

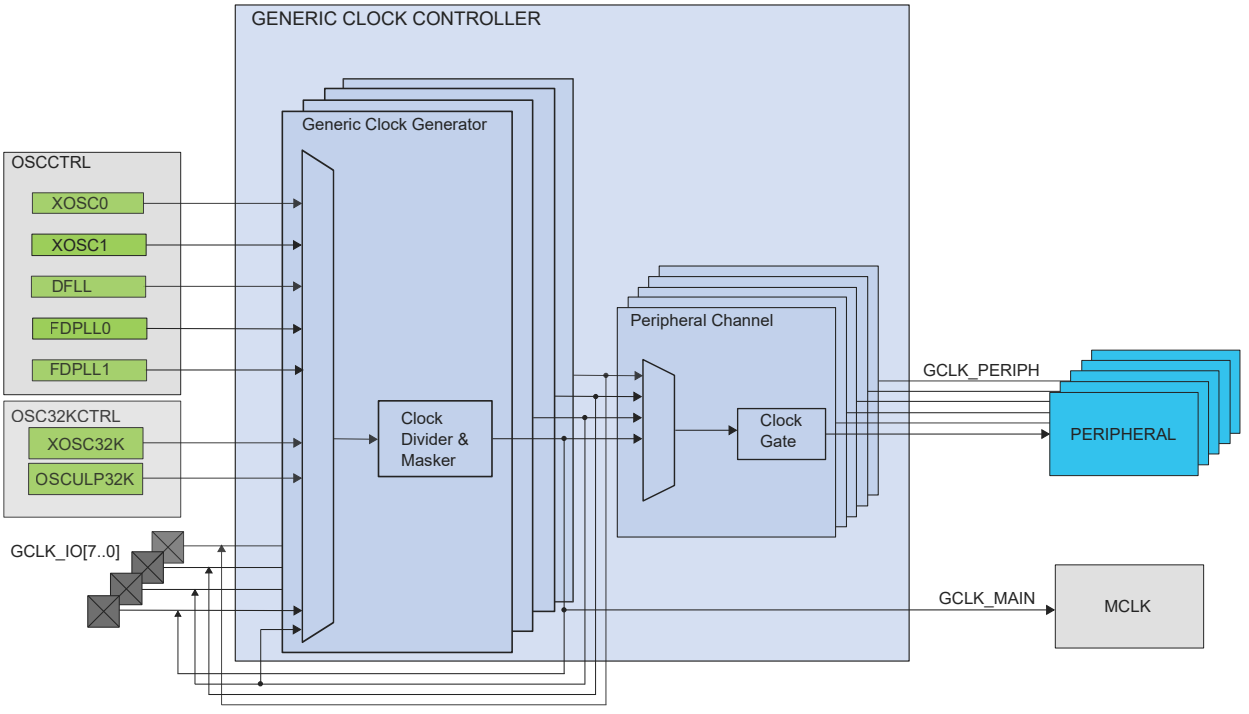
## 14.2. Features

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range:
  - Various clock sources
  - Embedded dividers

## 14.3. Block Diagram

The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in Device Clocking Diagram.

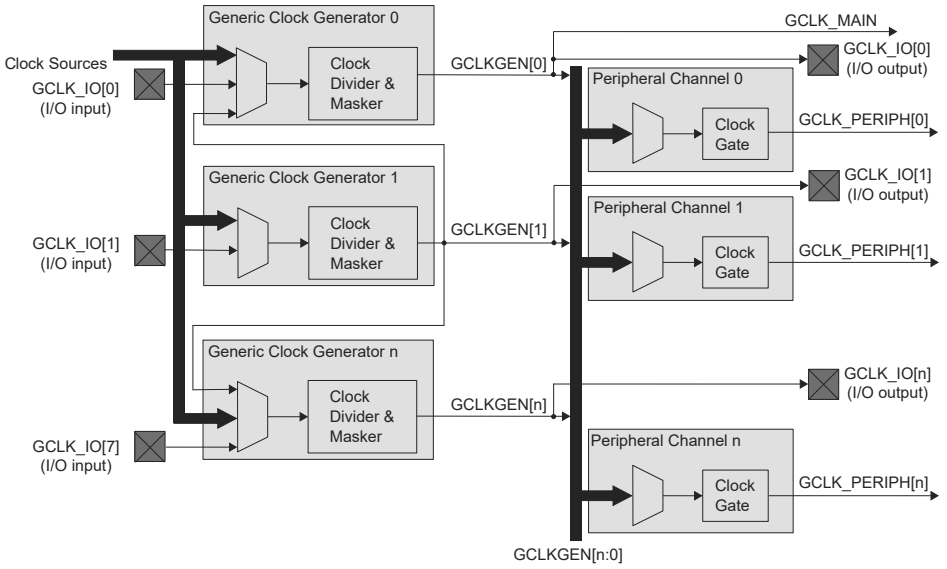
**Figure 14-1.** Device Clocking Diagram




The GCLK block diagram is shown below:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 14-2.** Generic Clock Controller Block Diagram




## 14.4. Signal Description

**Table 14-1.** GCLK Signal Description

| Signal Name | Type | Description |
|---|---|---|
| GCLK_IO[7:0] | Digital I/O | Clock source for Generators when input<br>Generic Clock signal when output |

**Note:** One signal can be mapped on several pins.

**Related Links**
I/O Multiplexing and Considerations

## 14.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 14.5.1. I/O Lines

Using the GCLK I/O lines requires the I/O pins to be configured.

**Related Links**
PORT - I/O Pin Controller

### 14.5.2. Power Management

The GCLK can operate in sleep modes, if required. Refer to the Sleep mode description in the Power Manager (PM) section.

**Related Links**
PM – Power Manager

### 14.5.3. Clocks

The GCLK bus clock (CLK_GCLK_APB) can be enabled and disabled in the Main Clock Controller.

**Related Links**
Peripheral Clock Masking
OSC32KCTRL – 32KHz Oscillators Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 14.5.4. DMA

Not applicable.

### 14.5.5. Interrupts

Not applicable.

### 14.5.6. Events

Not applicable.

### 14.5.7. Debug Operation

When the CPU is halted in debug mode the GCLK continues normal operation. If the GCLK is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

### 14.5.8. Register Access Protection

All registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC).

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

**Related Links**

PAC - Peripheral Access Controller

### 14.5.9. Analog Connections

Not applicable.

## 14.6. Functional Description

### 14.6.1. Principle of Operation

The GCLK module is comprised of twelve Generic Clock Generators (Generators) sourcing up to 48 Peripheral Channels and the Main Clock signal CLK_MAIN.

A clock source selected as input to a Generator can either be used directly, or it can be prescaled in the Generator. A generator output is used by one or more Peripheral Channels to provide a peripheral generic clock signal (GCLK_PERIPH[m]) to the peripherals.

### 14.6.2. Basic Operation

#### 14.6.2.1. Initialization

Before a Generator is enabled, the corresponding clock source should be enabled. The Peripheral clock must be configured as outlined by the following steps:

1. The Generator must be enabled (GENCTRLn.GENEN=1) and the division factor must be set (GENTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).
2. The Generic Clock for a peripheral must be configured by writing to the Peripheral Channel Control register (PCHCTRLm) belonging to the peripheral. The Generator used as the source for the Peripheral Clock must be written to the GEN bit field in the Peripheral Channel Control register (PCHCTRLm.GEN).

**Note:** Each Generator n is configured by one dedicated register GENCTRLn.

**Note:** Each Peripheral Channel m is configured by one dedicated register PCHCTRLm.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 14.6.2.2. Enabling, Disabling, and Resetting

The GCLK module has no enable/disable bit to enable or disable the whole module.

The GCLK is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST) to 1. All registers in the GCLK will be reset to their initial state, except for Peripheral Channels and associated Generators that have their Write Lock bit set to 1 (PCHCTRLm.WRTLOCK). For further details, refer to Configuration Lock.

### 14.6.2.3. Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of eight different clock sources except GCLK_GEN[1], which can be set to run from one of seven sources. GCLK_GEN[1] is the only Generator that can be selected as source to others Generators.

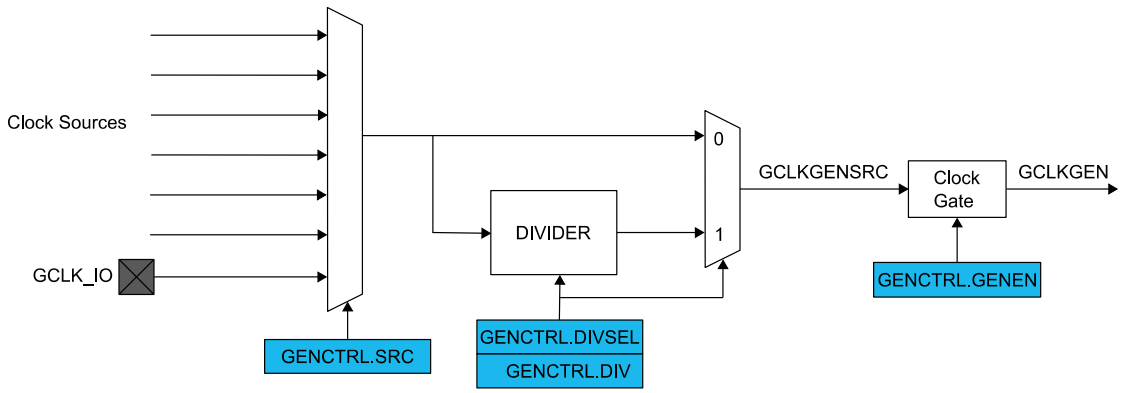
Each generator GCLK_GEN[x] can be connected to one specific pin GCLK_IO[x]. A pin GCLK_IO[x] can be set either to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as clock source for one or more Peripheral Channels. Each Generator output is allocated to one or several Peripherals.

GCLK_GEN[0] is used as GCLK_MAIN for the synchronous clock controller inside the Main Clock Controller. Refer to the Main Clock Controller description for details on the synchronous clock generation.

**Figure 14-3.** Generic Clock Generator




**Related Links**

MCLK – Main Clock

### 14.6.2.4. Enabling a Generator

A Generator is enabled by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN=1).

### 14.6.2.5. Disabling a Generator

A Generator is disabled by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN=0, the GCLK_GEN[n] clock is disabled and gated.

### 14.6.2.6. Selecting a Clock Source for the Generator

Each Generator can individually select a clock source by setting the Source Select bit group in the Generator Control register (GENCTRLn.SRC).

Changing from one clock source, for example A, to another clock source, B, can be done on the fly: If clock source B is not ready, the Generator will continue using clock source A. As soon as source B is


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

ready, the Generator will switch to it. During the switching operation, the Generator maintains clock requests to both clock sources A and B, and will release source A as soon as the switch is done. The according bit in SYNCBUSY register (SYNCBUSY.GENCTRLn) will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, DPLL). Only Generator 1 can be used as a common source for all other generators.

#### 14.6.2.7. Changing the Clock Frequency

The selected source for a Generator can be divided by writing a division value in the Division Factor bit field of the Generator Control register (GENCTRLn.DIV). How the actual division factor is calculated is depending on the Divide Selection bit (GENCTRLn.DIVSEL).

If GENCTRLn.DIVSEL=0 and GENCTRLn.DIV is either 0 or 1, the output clock will be undivided.

**Note:** The number of available DIV bits may vary from Generator to Generator.

#### 14.6.2.8. Duty Cycle

When dividing a clock with an odd division factor, the duty-cycle will not be 50/50. Setting the Improve Duty Cycle bit of the Generator Control register (GENCTRLn.IDC) will result in a 50/50 duty cycle.

#### 14.6.2.9. External Clock

The output clock (GCLK_GEN) of each Generator can be sent to I/O pins (GCLK_IO).

If the Output Enable bit in the Generator Control register is set (GENCTRLn.OE = 1) and the generator is enabled (GENCTRLn.GENEN=1), the Generator requests its clock source and the GCLK_GEN clock is output to an I/O pin.
**Note:** The I/O pin (GCLK/IO[n]) must first be configured as output by writing the corresponding PORT registers.

If GENCTRLn.OE is 0, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV: If GENCTRLn.OOV is '0', the output clock will be low. If this bit is '1', the output clock will be high.

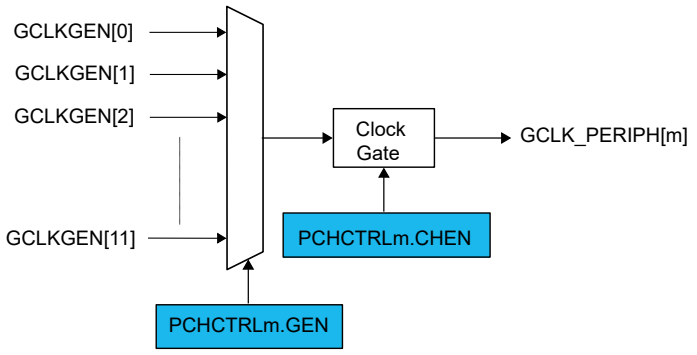
In Standby mode, if the clock is output (GENCTRLn.OE=1), the clock on the I/O pin is frozen to the OOV value if the Run In Standby bit of the Generic Control register (GENCTRLn.RUNSTDBY) is zero. If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.

**Related Links**

Power Domain Controller

### 14.6.3. Peripheral Clock

**Figure 14-4.** Peripheral Clock





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 14.6.3.1. Enabling a Peripheral Clock

Before a Peripheral Clock is enabled, one of the Generators must be enabled (GENCTRLn.GENEN) and selected as source for the Peripheral Channel by setting the Generator Selection bits in the Peripheral Channel Control register (PCHCTRLm.GEN). Any available Generator can be selected as clock source for each Peripheral Channel (See 'PCHCTRLm Mapping' for the mapping of peripheral to index m.).

When a Generator has been selected, the peripheral clock is enabled by setting the Channel Enable bit in the Peripheral Channel Control register, PCHCTRLm.CHEN = 1. The PCHCTRLm.CHEN bit must be synchronized to the generic clock domain. PCHCTRLm.CHEN will continue to read as its previous state until the synchronization is complete.

#### 14.6.3.2. Disabling a Peripheral Clock

A Peripheral Clock is disabled by writing PCHCTRLm.CHEN=0. The PCHCTRLm.CHEN bit must be synchronized to the Generic Clock domain. PCHCTRLm.CHEN will stay in its previous state until the synchronization is complete. The Peripheral Clock is gated when disabled.

**Related Links**

PCHCTRLm

Peripheral Channel Control

#### 14.6.3.3. Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to the PCHCTRLm.GEN bit, the peripheral clock must be disabled before re-enabling it with the new clock source setting. This prevents glitches during the transition:

1. Disable the Peripheral Channel by writing PCHCTRLm.CHEN = 0.
2. Verify that PCHCTRLm.CHEN reads '0'.
3. Change the source of the Peripheral Channel by writing PCHCTRLm.GEN.
4. Re-enable the Peripheral Channel by writing PCHCTRLm.CHEN = 1.
5. Assert that PCHCTRLm.CHEN reads '1'.

**Related Links**

PCHCTRLm

Peripheral Channel Control

#### 14.6.3.4. Configuration Lock

The peripheral clock configuration can be locked for further write accesses by setting the Write Lock bit in the Peripheral Channel Control register PCHCTRLm.WRTLOCK=1). All writing to the PCHCTRLm register will be ignored. It can only be unlocked by a Power Reset.

The Generator source of a locked Peripheral Channel will be locked, too: The corresponding GENCTRLn register is locked, and can be unlocked only by a Power Reset.

Regarding the configuration lock, there is one exception concerning the Generator 0. As it is used as GCLK_MAIN, it cannot be locked. It is reset by any Reset and will start up in a known configuration. The software reset (CTRLA.SWRST) can not unlock the registers.

In case of an external Reset (RESETN grounded), the Generator source will be disabled. Even if the WRTLOCK bit is written to '1' the peripheral channels are disabled (PCHCTRLm.CHEN set to '0') until the Generator source is enabled again. Then, the PCHCTRLm.CHEN are set to '1' again.

**Related Links**

CTRLA

Control A


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 14.6.4. Additional Features

### 14.6.4.1. Peripheral Clock Enable after Reset

The Generic Clock Controller must be able to provide a generic clock to some specific peripherals after a Reset. That means that the configuration of the Generators and Peripheral Channels after Reset is device-dependent.

Refer to GENCTRLn.SRC for details on GENCTRLn reset.

Refer to PCHCTRLm.SRC for details on PCHCTRLm reset.

## 14.6.5. Sleep Mode Operation

### 14.6.5.1. SleepWalking

The GCLK module supports the SleepWalking feature.

If the system is in a sleep mode where the Generic Clocks are stopped, a peripheral that needs its clock in order to execute a process must first request it from the Generic Clock Controller.

The Generic Clock Controller receives this request, determines which Generic Clock Generator is involved and which clock source needs to be awakened. It then wakes up the respective clock source, enables the Generator and Peripheral Channel stages successively, and delivers the clock to the peripheral.

The RUNSTDBY bit in the Generator Control register controls clock output to pin during standby sleep mode. If the bit is cleared, the Generator output is not available on a pin. When set, the GCLK can continuously output the generator output to GCLK_IO[n]. Refer to a External Clock for details.

**Related Links**

PM – Power Manager

### 14.6.5.2. Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby Mode, minimizing the power consumption:

**Table 14-2.** Clock Generator n Activity in Standby Mode

| Request for Clock n present | GENCTRLn.RUNSTDBY | GENCTRLn.OE | Clock Generator n |
|---|---|---|---|
| yes | - | - | active |
| no | 1 | 1 | active |
| no | 1 | 0 | OFF |
| no | 0 | 1 | OFF |
| no | 0 | 0 | OFF |

### 14.6.5.3. Entering Standby Mode

There may occur a delay when the device is put into Standby before the power is turned off. This delay is caused by running Clock Generators: if the Run in Standby bit in the Generator Control register (GENCTRLn.RUNSTDBY) is '0', GCLK must verify that the clock is turned of properly. The duration of this verification is frequency-dependent.

**Related Links**

PM – Power Manager

## 14.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- Generic Clock Generator Control register (GENCTRLn)
- Control A register (CTRLA)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

**Related Links**

CTRLA

Control A

PCHCTRLm

Peripheral Channel Control


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 14.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | | | SWRST |
| 0x01<br>...<br>0x03 | Reserved | | | | | | | | | |
| 0x04 | SYNCBUSY | 7:0 | GENCTRL5 | GENCTRL4 | GENCTRL3 | GENCTRL2 | GENCTRL1 | GENCTRL0 | | SWRST |
| | | 15:8 | | | GENCTRL11 | GENCTRL10 | GENCTRL9 | GENCTRL8 | GENCTRL7 | GENCTRL6 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08<br>...<br>0x1F | Reserved | | | | | | | | | |
| 0x20 | GENCTRL0 | 7:0 | | | | | SRC[3:0] | | | |
| | | 15:8 | | | RUNSTDBY | DIVSEL | OE | OOV | IDC | GENEN |
| | | 23:16 | DIV[7:0] | | | | | | | |
| | | 31:24 | DIV[15:8] | | | | | | | |
| ... | | | | | | | | | | |
| 0x4C | GENCTRL11 | 7:0 | | | | | SRC[3:0] | | | |
| | | 15:8 | | | RUNSTDBY | DIVSEL | OE | OOV | IDC | GENEN |
| | | 23:16 | DIV[7:0] | | | | | | | |
| | | 31:24 | DIV[15:8] | | | | | | | |
| 0x50<br>...<br>0x7F | Reserved | | | | | | | | | |
| 0x80 | PCHCTRL0 | 7:0 | WRTLOCK | CHEN | | | GEN[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| ... | | | | | | | | | | |
| 0x013C | PCHCTRL47 | 7:0 | WRTLOCK | CHEN | | | GEN[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 14.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 14.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SWRST |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – SWRST** Software Reset
Writing a zero to this bit has no effect.
Setting this bit to 1 will reset all registers in the GCLK to their initial state after a Power Reset, except for generic clocks and associated Generators that have their WRTLOCK bit in PCHCTRLm set to 1.
Refer to GENCTRL Reset Value for details on GENCTRL register reset.
Refer to PCHCTRL Reset Value for details on PCHCTRL register reset.
Due to synchronization, there is a waiting period between setting CTRLA.SWRST and a completed Reset. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

| Value | Description |
|---|---|
| 0 | There is no Reset operation ongoing. |
| 1 | A Reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 14.8.2. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | GENCTRL11 | GENCTRL10 | GENCTRL9 | GENCTRL8 | GENCTRL7 | GENCTRL6 |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GENCTRL5 | GENCTRL4 | GENCTRL3 | GENCTRL2 | GENCTRL1 | GENCTRL0 | | SWRST |
| Access | R | R | R | R | R | R | | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |

**Bits 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 – GENCTRLn** Generator Control n Synchronization Busy
This bit is cleared when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is complete, or when clock switching operation is complete.
This bit is set when the synchronization of the Generator Control n register (GENCTRLn) between clock domains is started.

**Bit 0 – SWRST** Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST register bit between clock domains is complete.
This bit is set when the synchronization of the CTRLA.SWRST register bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 14.8.3. Generator Control

**Name:** GENCTRLn
**Offset:** 0x20 + n*0x04 [n=0..11]
**Reset:** 0x00000106
**Property:** PAC Write-Protection, Write-Synchronized

GENCTRLn controls the settings of Generic Generator n (n=[11:0]). The reset value is 0x00000106 for Generator n=0, else 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIV[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DIV[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | RUNSTDBY | DIVSEL | OE | OOV | IDC | GENEN |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SRC[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 31:16 – DIV[15:0]** Division Factor
These bits represent a division value for the corresponding Generator. The actual division factor is dependent on the state of DIVSEL. The number of relevant DIV bits for each Generator can be seen in this table. Written bits outside of the specified range will be ignored.

**Table 14-3.** Division Factor Bits

| Generic Clock Generator | Division Factor Bits | Maximum Division Factor |
|---|---|---|
| Generator 0 | 8 division factor bits - DIV[7:0] | 512 |
| Generator 1 | 16 division factor bits - DIV[15:0] | 131072 |
| Generator 2 - 11 | 8 division factor bits - DIV[7:0] | 512 |

**Bit 13 – RUNSTDBY** Run in Standby
This bit is used to keep the Generator running in Standby as long as it is configured to output to a dedicated GCLK_IOn pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.

| Value | Description |
|---|---|
| 0 | The Generator is stopped in Standby and the GCLK_IOn pin state (one or zero) will be dependent on the setting in GENCTRL.OOV. |
| 1 | The Generator is kept running and output to its dedicated GCLK_IOn pin during Standby mode. |

**Bit 12 – DIVSEL** Divide Selection
This bit determines how the division factor of the clock source of the Generator will be calculated from DIV. If the clock source should not be divided, DIVSEL must be 0 and the GENCTRLn.DIV value must be either 0 or 1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The Generator clock frequency equals the clock source frequency divided by GENCTRLn.DIV. |
| 1 | The Generator clock frequency equals the clock source frequency divided by 2^(N+1), where N is the Division Factor Bits for the selected generator (refer to GENCTRLn.DIV). |

**Bit 11 – OE** Output Enable

This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO[7..0]), as long as GCLK_IOn is not defined as the Generator source in the GENCTRLn.SRC bit field.

| Value | Description |
|---|---|
| 0 | No Generator clock signal on pin GCLK_IOn. |
| 1 | The Generator clock signal is output on the corresponding GCLK_IOn, unless GCLK_IOn is selected as a generator source in the GENCTRLn.SRC bit field. |

**Bit 10 – OOV** Output Off Value

This bit is used to control the clock output value on pin (GCLK_IO[7..0]) when the Generator is turned off or the OE bit is zero, as long as GCLK_IOn is not defined as the Generator source in the GENCTRLn.SRC bit field.

| Value | Description |
|---|---|
| 0 | The GCLK_IOn will be LOW when generator is turned off or when the OE bit is zero. |
| 1 | The GCLK_IOn will be HIGH when generator is turned off or when the OE bit is zero. |

**Bit 9 – IDC** Improve Duty Cycle

This bit is used to improve the duty cycle of the Generator output to 50/50 for odd division factors.

| Value | Description |
|---|---|
| 0 | Generator output clock duty cycle is not balanced to 50/50 for odd division factors. |
| 1 | Generator output clock duty cycle is 50/50. |

**Bit 8 – GENEN** Generator Enable

This bit is used to enable and disable the Generator.

| Value | Description |
|---|---|
| 0 | Generator is disabled. |
| 1 | Generator is enabled. |

**Bits 3:0 – SRC[3:0]** Generator Clock Source Selection

These bits select the Generator clock source, as shown in this table.

**Table 14-4.** Generator Clock Source Selection

| Value | Name | Description |
|---|---|---|
| 0x00 | XOSC0 | XOSC 0 oscillator output |
| 0x01 | XOSC1 | XOSC 1 oscillator output |
| 0x02 | GCLK_IN | Generator input pad (GCLK_IO) |
| 0x03 | GCLK_GEN1 | Generic clock generator 1 output |
| 0x04 | OSCULP32K | OSCULP32K oscillator output |
| 0x05 | XOSC32K | XOSC32K oscillator output |
| 0x06 | DFLL | DFLL48M oscillator output |
| 0x07 | DPLL0 | FDPLL200M0 output |
| 0x08 | DPLL1 | FDPLL200M1 output |
| 0x09-0x0F | Reserved | Reserved |

A Power Reset will reset all GENCTRLn registers. the Reset values of the GENCTRLn registers are shown in table below.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 14-5.** GENCTRLn Reset Value after a Power Reset

| GCLK Generator | Reset Value after a Power Reset |
|---|---|
| 0 | 0x00000106 |
| others | 0x00000000 |

A User Reset will reset the associated GENCTRL register unless the Generator is the source of a locked Peripheral Channel (PCHCTRLm.WRTLOCK=1). The reset values of the GENCTRL register are as shown in the table below.

**Table 14-6.** GENCTRLn Reset Value after a User Reset

| GCLK Generator | Reset Value after a User Reset |
|---|---|
| 0 | 0x00000106 |
| others | No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK=1<br>else 0x00000000 |

### Related Links

PCHCTRLm

Peripheral Channel Control


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 14.8.4. Peripheral Channel Control

**Name:** PCHCTRLm
**Offset:** 0x80 + m*0x04 [m=0..47]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

PCHTRLm controls the settings of Peripheral Channel number m (m=[47:0]).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WRTLOCK | CHEN | | | GEN[3:0] | | | |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

**Bit 7 – WRTLOCK** Write Lock
After this bit is set to '1', further writes to the PCHCTRLm register will be discarded. The control register of the corresponding Generator n (GENCTRLn), as assigned in PCHCTRLm.GEN, will also be locked. It can only be unlocked by a Power Reset.
Note that Generator 0 cannot be locked.

| Value | Description |
|---|---|
| 0 | The Peripheral Channel register and the associated Generator register are not locked |
| 1 | The Peripheral Channel register and the associated Generator register are locked |

**Bit 6 – CHEN** Channel Enable
This bit is used to enable and disable a Peripheral Channel.

| Value | Description |
|---|---|
| 0 | The Peripheral Channel is disabled |
| 1 | The Peripheral Channel is enabled |

**Bits 3:0 – GEN[3:0]** Generator Selection
This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:

**Table 14-7.** Generator Selection

| Value | Description |
|---|---|
| 0x0 | Generic Clock Generator 0 |
| 0x1 | Generic Clock Generator 1 |
| 0x2 | Generic Clock Generator 2 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 14-7.** Generator Selection (continued)

| Value | Description |
|---|---|
| 0x3 | Generic Clock Generator 3 |
| 0x4 | Generic Clock Generator 4 |
| 0x5 | Generic Clock Generator 5 |
| 0x6 | Generic Clock Generator 6 |
| 0x7 | Generic Clock Generator 7 |
| 0x8 | Generic Clock Generator 8 |
| 0x9 | Generic Clock Generator 9 |
| 0xA | Generic Clock Generator 10 |
| 0xB | Generic Clock Generator 11 |

**Table 14-8.** Reset Value after a User Reset or a Power Reset

| Reset | PCHCTRLm.GEN | PCHCTRLm.CHEN | PCHCTRLm.WRTLOCK |
|---|---|---|---|
| Power Reset | 0x0 | 0x0 | 0x0 |
| User Reset | If WRTLOCK = 0<br>: 0x0<br>If WRTLOCK = 1: no change | If WRTLOCK = 0<br>: 0x0<br>If WRTLOCK = 1: no change | No change |

A Power Reset will reset all the PCHCTRLm registers.
A User Reset will reset a PCHCTRL if WRTLOCK=0, or else, the content of that PCHCTRL remains unchanged.
The PCHCTRL register Reset values are shown in the table below, PCHCTRLm Mapping.

**Table 14-9.** PCHCTRLm Mapping

| index(m) | Name | Description |
|---|---|---|
| 0 | GCLK_OSCCTRL_DFLL48 | DFLL48 input clock source |
| 1 | GCLK_OSCCTRL_FDPLL0 | Reference clock for FDPLL0 |
| 2 | GCLK_OSCCTRL_FDPLL1 | Reference clock for FDPLL1 |
| 3 | GCLK_OSCCTRL_FDPLL0_32K<br>GCLK_OSCCTRL_FDPLL1_32K<br>GCLK_SDHC0_SLOW<br>GCLK_SDHC1_SLOW<br>GCLK_SERCOM[0..7]_SLOW | FDPLL0 32KHz clock for internal lock timer<br>FDPLL1 32KHz clock for internal lock timer<br>SDHC0 Slow<br>SDHC1 Slow<br>SERCOM[0..7] Slow |
| 4 | GCLK_EIC | EIC |
| 5 | GCLK_FREQM_MSR | FREQM Measure |
| 6 | GCLK_FREQM_REF | FREQM Reference |
| 7 | GCLK_SERCOM0_CORE | SERCOM0 Core |
| 8 | GCLK_SERCOM1_CORE | SERCOM1 Core |
| 9 | GCLK_TC0, GCLK_TC1 | TC0, TC1 |
| 10 | GCLK_USB | USB |
| 22:11 | GCLK_EVSYS[0..11] | EVSYS[0..11] |
| 23 | GCLK_SERCOM2_CORE | SERCOM2 Core |
| 24 | GCLK_SERCOM3_CORE | SERCOM3 Core |
| 25 | GCLK_TCC0, GCLK_TCC1 | TCC0, TCC1 |
| 26 | GCLK_TC2, GCLK_TC3 | TC2, TC3 |
| 27 | GCLK_CAN0 | CAN0 |
| 28 | GCLK_CAN1 | CAN1 |
| 29 | GCLK_TCC2, GCLK_TCC3 | TCC2, TCC3 |
| 30 | GCLK_TC4, GCLK_TC5 | TC4, TC5 |
| 31 | GCLK_PDEC | PDEC |
| 32 | GCLK_AC | AC |
| 33 | GCLK_CCL | CCL |
| 34 | GCLK_SERCOM4_CORE | SERCOM4 Core |
| 35 | GCLK_SERCOM5_CORE | SERCOM5 Core |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 14-9.** PCHCTRLm Mapping (continued)

| index(m) | Name | Description |
|---|---|---|
| 36 | GCLK_SERCOM6_CORE | SERCOM6 Core |
| 37 | GCLK_SERCOM7_CORE | SERCOM7 Core |
| 38 | GCLK_TCC4 | TCC4 |
| 39 | GCLK_TC6, GCLK_TC7 | TC6, TC7 |
| 40 | GCLK_ADC0 | ADC0 |
| 41 | GCLK_ADC1 | ADC1 |
| 42 | GCLK_DAC | DAC |
| 44:43 | GCLK_I2S | I2S |
| 45 | GCLK_SDHC0 | SDHC0 |
| 46 | GCLK_SDHC1 | SDHC1 |
| 47 | GCLK_CM4_TRACE | CM4 Trace |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 15. MCLK – Main Clock

## 15.1. Overview

The Main Clock (MCLK) controls the synchronous clock generation of the device.

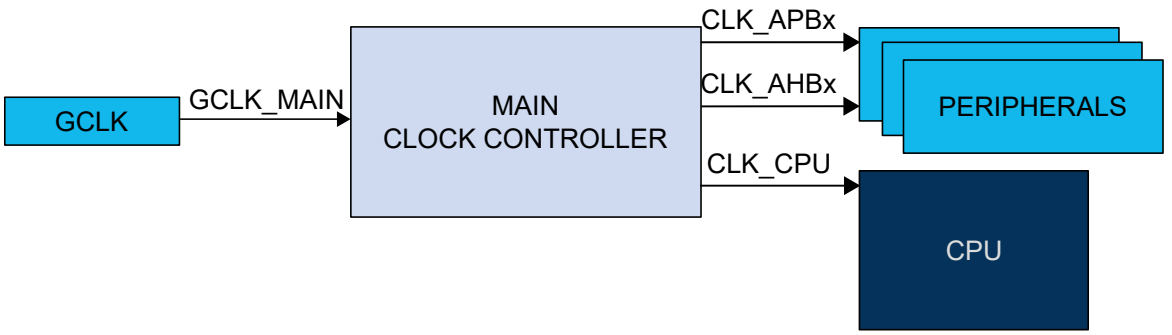
Using a clock provided by the Generic Clock Module (GCLK_MAIN), the Main Clock Controller provides synchronous system clocks to the CPU and the modules connected to the AHBx and the APBx buses. The synchronous system clocks are divided into a number of clock domains. Each clock domain can run at different frequencies, enabling the user to save power by running peripherals at a relatively low clock frequency, while maintaining high CPU performance or vice versa. In addition, the clock can be masked for individual modules, enabling the user to minimize power consumption by disabling individual modules.

## 15.2. Features

- Generates CPU, AHB, and APB system clocks
  - Clock source and division factor from GCLK
  - Clock prescaler with 1x to 128x division
- Safe run-time clock switching from GCLK
- Module-level clock gating through maskable peripheral clocks

## 15.3. Block Diagram

**Figure 15-1.** MCLK Block Diagram




x = A, B, C or D

## 15.4. Signal Description

Not applicable.

## 15.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 15.5.1. I/O Lines

Not applicable.

### 15.5.2. Power Management

The MCLK will operate in all sleep modes if a synchronous clock is required in these modes.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
PM – Power Manager

### 15.5.3. Clocks

The MCLK bus clock (CLK_MCLK_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_MCLK_APB can be found in the Peripheral Clock Masking section. If this clock is disabled, it can only be re-enabled by a reset.

The Generic Clock GCLK_MAIN is required to generate the Main Clocks. GCLK_MAIN is configured in the Generic Clock Controller, and can be re-configured by the user if needed.

**Related Links**
GCLK - Generic Clock Controller

#### 15.5.3.1. Main Clock

The main clock CLK_MAIN is the common source for the synchronous clocks. This is fed into the common 8-bit prescaler that is used to generate synchronous clocks to the CPU, AHBx, and APBx modules.

#### 15.5.3.2. CPU Clock

The CPU clock (CLK_CPU) is routed to the CPU. Halting the CPU clock inhibits the CPU from executing instructions.

#### 15.5.3.3. APBx and AHBx Clock

The APBx clocks (CLK_APBx) and the AHBx clocks (CLK_AHBx) are the root clock source used by modules requiring a clock on the APBx and the AHBx bus. These clocks are always synchronous to the CPU clock, and can run even when the CPU clock is turned off in sleep mode. A clock gater is inserted after the common APB clock to gate any APBx clock of a module on APBx bus, as well as the AHBx clock.

#### 15.5.3.4. Clock Domains

The device has these synchronous clock domains:

- High-Speed synchronous clock domain (HS Clock Domain). Frequency is $f_{HS}$.
- CPU synchronous clock domain (CPU Clock Domain). Frequency is $f_{CPU}$.

See also the related links for the clock domain partitioning.

### 15.5.4. DMA

Not applicable.

### 15.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the MCLK interrupt requires the Interrupt Controller to be configured first.

### 15.5.6. Events

Not applicable.

### 15.5.7. Debug Operation

When the CPU is halted in debug mode, the MCLK continues normal operation. In sleep mode, the clocks generated from the MCLK are kept running to allow the debugger accessing any module. As a consequence, power measurements are incorrect in debug mode.

### 15.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Interrupt Flag register (INTFLAG)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**

PAC - Peripheral Access Controller

### 15.5.9. Analog Connections

Not applicable.

## 15.6. Functional Description

### 15.6.1. Principle of Operation

The CLK_MAIN clock signal from the GCLK module is the source for the main clock, which in turn is the common root for the synchronous clocks for the CPU, APBx, and AHBx modules. The CLK_MAIN is divided by an 8-bit prescaler. Each of the derived clocks can run from any divided or undivided main clock, ensuring synchronous clock sources for each clock domain. The clock domain (CPU) can be changed on the fly to respond to variable load in the application. The clocks for each module in a clock domain can be masked individually to avoid power consumption in inactive modules. Depending on the sleep mode, some clock domains can be turned off.

### 15.6.2. Basic Operation

#### 15.6.2.1. Initialization

After a Reset, the default clock source of the CLK_MAIN clock (GCLK_MAIN) is started and calibrated before the CPU starts running. The GCLK_MAIN clock is selected as the main clock without any prescaler division.

By default, only the necessary clocks are enabled.

#### 15.6.2.2. Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

#### 15.6.2.3. Selecting the Main Clock Source

Refer to the Generic Clock Controller description for details on how to configure the clock source of the GCLK_MAIN clock.

**Related Links**

GCLK - Generic Clock Controller

#### 15.6.2.4. Selecting the Synchronous Clock Division Ratio

The main clock GCLK_MAIN feeds an 8-bit prescaler, which can be used to generate the synchronous clocks. By default, the synchronous clocks run on the undivided main clock. The user can select a prescaler division for the CPU clock domain by writing the Division (DIV) bits in the CPU Clock Division register CPUDIV, resulting in a CPU clock domain frequency determined by this equation:

$$f_{CPU} = \frac{f_{main}}{CPUDIV}$$

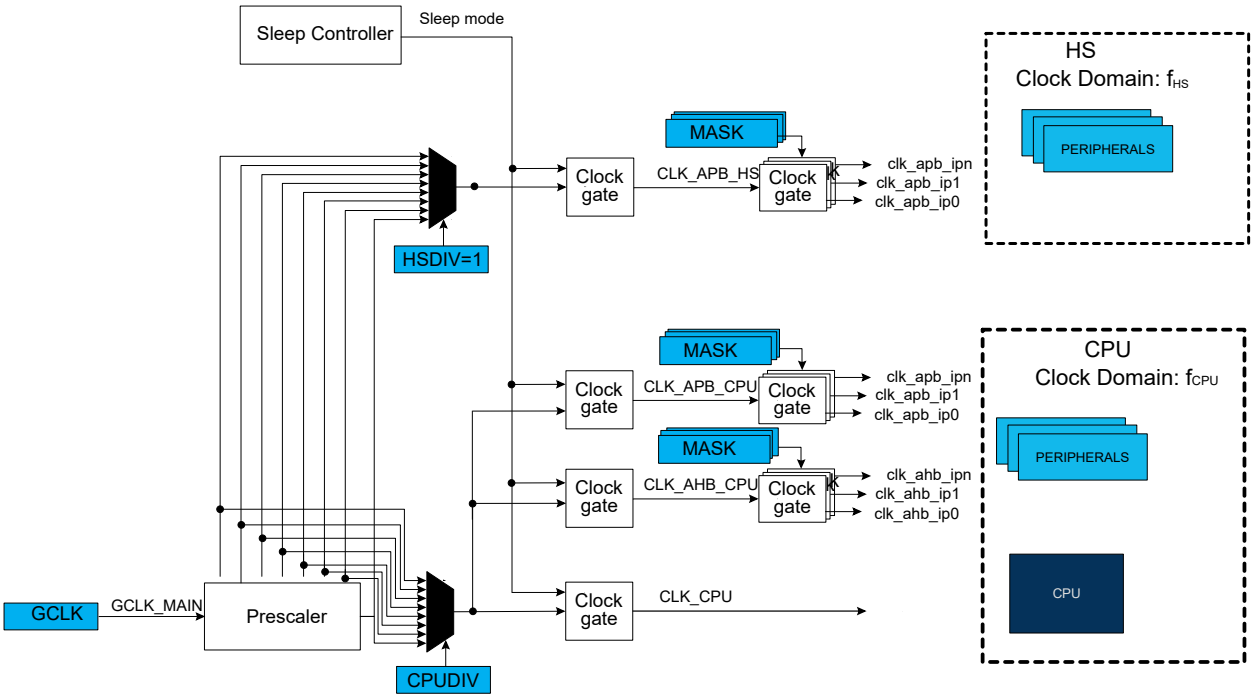
Frequencies must never exceed the specified maximum frequency for each clock domain given in the electrical characteristics specifications.

If the application attempts to write forbidden values in CPUDIV register, register is written but these bad values are not used and a violation is reported to the PAC module.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Division bits (DIV) can be written without halting or disabling peripheral modules. Writing DIV bits allows a new clock setting to be written to all synchronous clocks belonging to the corresponding clock domain at the same time.

**Figure 15-2.** Synchronous Clock Selection and Prescaler




**Note:** A FAST clock for QSPI (CLK_QSPI2X_AHB) is derived from high-speed synchronous $f_{HS}$.

**Related Links**

PAC - Peripheral Access Controller

### 15.6.2.5. Clock Ready Flag

There is a slight delay between writing to CPUDIV until the new clock settings become effective.

During this interval, the Clock Ready flag in the Interrupt Flag Status and Clear register (INTFLAG.CKRDY) will return zero when read. If CKRDY in the INTENSET register is set to '1', the Clock Ready interrupt will be triggered when the new clock setting is effective. The new CPUDIV value must not be re-written while INTFLAG.CKRDY reads '0'. The system may become unstable or hang, and a violation is reported to the PAC module.

**Related Links**

PAC - Peripheral Access Controller

### 15.6.2.6. Peripheral Clock Masking

The user can disable or enable the AHB or APB clock for a peripheral by writing the corresponding bit in the Clock Mask registers (APBxMASK) to '0' or '1'. The default state of the peripheral clocks is given in the table below:

**Table 15-1.** Peripheral Clock Default State

| CPU Clock Domain | |
|---|---|
| **Peripheral Clock** | **Default State** |
| CLK_AC_APB | Disabled |
| CLK_ADC0_APB | Disabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 15-1.** Peripheral Clock Default State (continued)

| CPU Clock Domain | |
|---|---|
| **Peripheral Clock** | **Default State** |
| CLK_ADC1_APB | Disabled |
| CLK_AES_APB | Disabled |
| CLK_BRIDGE_A_AHB | Enabled |
| CLK_BRIDGE_B_AHB | Enabled |
| CLK_BRIDGE_C_AHB | Enabled |
| CLK_BRIDGE_D_AHB | Enabled |
| CLK_CAN0_AHB | Enabled |
| CLK_CAN1_AHB | Enabled |
| CLK_CMCC_AHB | Enabled |
| CLK_DMAC_AHB | Enabled |
| CLK_DSU_AHB | Enabled |
| CLK_EIC_APB | Enabled |
| CLK_EVSYS_APB | Disabled |
| CLK_FREQM_APB | Disabled |
| CLK_GCLK_APB | Enabled |
| CLK_GMAC_AHB | Enabled |
| CLK_GMAC_APB | Disabled |
| CLK_ICM_AHB | Enabled |
| CLK_I2S_AHB | Disabled |
| CLK_MCLK_APB | Enabled |
| CLK_NVMCTRL_AHB | Enabled |
| CLK_NVMCTRL_APB | Enabled |
| CLK_NVMCTRL_CACHE | Enabled |
| CLK_NVMCTRL_SMEEPROM | Enabled |
| CLK_OSCCTRL_APB | Enabled |
| CLK_PAC_AHB | Enabled |
| CLK_PAC_APB | Enabled |
| CLK_PDEC_APB | Disabled |
| CLK_PORT_APB | Enabled |
| CLK_PUKCC_AHB | Enabled |
| CLK_QSPI_AHB | Enabled |
| CLK_QSPI2X_AHB | Enabled |
| CLK_SDHC0_AHB | Enabled |
| CLK_SDHC1_AHB | Enabled |
| CLK_SERCOM0_APB | Disabled |
| CLK_SERCOM1_APB | Disabled |
| CLK_SERCOM2_APB | Disabled |
| CLK_SERCOM3_APB | Disabled |
| CLK_SERCOM4_APB | Disabled |
| CLK_SERCOM5_APB | Disabled |
| CLK_SERCOM6_APB | Disabled |
| CLK_SERCOM7_APB | Disabled |
| CLK_TC0_APB | Disabled |
| CLK_TC1_APB | Disabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 15-1.** Peripheral Clock Default State (continued)

| CPU Clock Domain | |
|---|---|
| **Peripheral Clock** | **Default State** |
| CLK_TC2_APB | Disabled |
| CLK_TC3_APB | Disabled |
| CLK_TC4_APB | Disabled |
| CLK_TC5_APB | Disabled |
| CLK_TC6_APB | Disabled |
| CLK_TC7_APB | Disabled |
| CLK_TCC0_APB | Disabled |
| CLK_TCC1_APB | Disabled |
| CLK_TCC2_APB | Disabled |
| CLK_TCC3_APB | Disabled |
| CLK_TCC4_APB | Disabled |
| CLK_USB_AHB | Enabled |
| CLK_USB_APB | Disabled |
| CLK_WDT_APB | Enabled |
| CLK_DAC_APB | Disabled |
| CLK_DSU_APB | Enabled |
| CLK_CCL_APB | Disabled |
| CLK_QSPI_APB | Enabled |
| CLK_ICM_APB | Disabled |
| CLK_TRNG_APB | Disabled |
| **Backup Clock Domain** | |
| **Peripheral Clock** | **Default State** |
| CLK_OSC32KCTRL_APB | Enabled |
| CLK_PM_APB | Enabled |
| CLK_SUPC_APB | Enabled |
| CLK_RSTC_APB | Enabled |
| CLK_RTC_APB | Enabled |

When the APB clock is not provided to a module, its registers cannot be read or written. The module can be re-enabled later by writing the corresponding mask bit to '1'.

A module may be connected to several clock domains (for example, AHB and APB), in which case it will have several mask bits.

The clocks should only be switched off if it is certain that the module will not be used: Switching off the clock for the NVM Controller (NVMCTRL) will cause a problem if the CPU needs to read from the Flash memory. Switching off the clock to the MCLK module (which contains the mask registers) or the corresponding APBx bridge, will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

### 15.6.3. DMA Operation

Not applicable.

### 15.6.4. Interrupts

The peripheral has the following interrupt sources:

- Clock Ready (CKRDY): indicates that CPU clocks are ready. This interrupt is a synchronous wake-up source.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be enabled individually by writing a '1' to the corresponding enabling bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding clearing bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset. An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources.If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.

**Related Links**

PM – Power Manager

Overview

### 15.6.5. Events

Not applicable.

### 15.6.6. Sleep Mode Operation

In IDLE sleep mode, the MCLK is still running on the selected main clock.

In STANDBY sleep mode, the MCLK is frozen if no synchronous clock is required.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 15.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | Reserved | | | | | | | | | |
| 0x01 | INTENCLR | 7:0 | | | | | | | | CKRDY |
| 0x02 | INTENSET | 7:0 | | | | | | | | CKRDY |
| 0x03 | INTFLAG | 7:0 | | | | | | | | CKRDY |
| 0x04 | HSDIV | 7:0 | DIV[7:0] | | | | | | | |
| 0x05 | CPUDIV | 7:0 | DIV[7:0] | | | | | | | |
| 0x06 ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | AHBMASK | 7:0 | Reserved | NVMCTRL | Reserved | DSU | HPB3 | HPB2 | HPB1 | HPB0 |
| | | 15:8 | SDHC0 | GMAC | QSPI | PAC | Reserved | USB | DMAC | CMCC |
| | | 23:16 | NVMCTRL_CACHE | NVMCTRL_SMEEPROM | QSPI_2X | PUKCC | ICM | CAN1 | CAN0 | SDHC1 |
| | | 31:24 | | | | | | | | |
| 0x14 | APBAMASK | 7:0 | GCLK | SUPC | OSC32KCTRL | OSCCTRL | RSTC | MCLK | PM | PAC |
| | | 15:8 | TC1 | TC0 | SERCOM1 | SERCOM0 | FREQM | EIC | RTC | WDT |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | APBBMASK | 7:0 | EVSYS | | | PORT | | NVMCTRL | DSU | USB |
| | | 15:8 | | TC3 | TC2 | TCC1 | TCC0 | SERCOM3 | SERCOM2 | |
| | | 23:16 | | | | | | | | RAMECC |
| | | 31:24 | | | | | | | | |
| 0x1C | APBCMASK | 7:0 | PDEC | TC5 | TC4 | TCC3 | TCC2 | GMAC | | |
| | | 15:8 | | CCL | QSPI | | ICM | TRNG | AES | AC |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | APBDMASK | 7:0 | ADC0 | TC7 | TC6 | TCC4 | SERCOM7 | SERCOM6 | SERCOM5 | SERCOM4 |
| | | 15:8 | | | | | PCC | I2S | DAC | ADC1 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 15.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers can be write-protected optionally by the Peripheral Access Controller (PAC). This is denoted by the property "PAC Write-Protection" in each individual register description. Refer to the Register Access Protection for details.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 15.8.1. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x01
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CKRDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – CKRDY** Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Clock Ready interrupt is enabled and will generate an interrupt request when the Clock Ready Interrupt Flag is set. |
| 1 | The Clock Ready interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 15.8.2. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x02
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CKRDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – CKRDY** Clock Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Clock Ready interrupt is disabled. |
| 1 | The Clock Ready interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 15.8.3. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x03
**Reset:** 0x01
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CKRDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 1 |

**Bit 0 – CKRDY** Clock Ready
This flag is cleared by writing a '1' to the flag.
This flag is set when the synchronous CPU, APBx, and AHBx clocks are stable and will generate an interrupt if INTENCLR/SET.CKRDY is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Clock Ready interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 15.8.4. High-Speed Clock Division

**Name:** HSDIV
**Offset:** 0x04
**Reset:** 0x01

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIV[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 7:0 – DIV[7:0]** HS Clock Division Factor
These bits define the division ratio of the main clock prescaler related to the HS clock domain (HSDIV).

| Value | Name | Description |
|---|---|---|
| 0x01 | DIV1 | Divide by 1 |
| others | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 15.8.5. CPU Clock Division

**Name:** CPUDIV
**Offset:** 0x05
**Reset:** 0x01
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIV[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 7:0 – DIV[7:0]** CPU Clock Division Factor
These bits define the division ratio of the main clock prescaler related to the CPU clock domain (CPUDIV).
To ensure correct operation, frequencies must be selected so that $f_{HS} \geq f_{CPU}$ (i.e. CPUDIV ≥ HSDIV). Frequencies must never exceed the specified maximum frequency for each clock domain.

| Value | Name | Description |
|---|---|---|
| 0x01 | DIV1 | Divide by 1 |
| 0x02 | DIV2 | Divide by 2 |
| 0x04 | DIV4 | Divide by 4 |
| 0x08 | DIV8 | Divide by 8 |
| 0x10 | DIV16 | Divide by 16 |
| 0x20 | DIV32 | Divide by 32 |
| 0x40 | DIV64 | Divide by 64 |
| 0x80 | DIV128 | Divide by 128 |
| others | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 15.8.6. AHB Mask

**Name:** AHBMASK
**Offset:** 0x10
**Reset:** 0x00FFFFFF
**Property:** PAC Write-Protection

**Note:** All "Reserved" bits should be set to 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NVMCTRL_CACHE | NVMCTRL_SMEEPROM | QSPI_2X | PUKCC | ICM | CAN1 | CAN0 | SDHC1 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SDHC0 | GMAC | QSPI | PAC | Reserved | USB | DMAC | CMCC |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Reserved | NVMCTRL | Reserved | DSU | HPB3 | HPB2 | HPB1 | HPB0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bit 23 – NVMCTRL_CACHE** NVMCTRL_CACHE AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the NVMCTRL_CACHE is stopped. |
| 1 | The AHB clock for the NVMCTRL_CACHE is enabled. |

**Bit 22 – NVMCTRL_SMEEPROM** NVMCTRL_SMEEPROM AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the NVMCTRL_SMEEPROM is stopped. |
| 1 | The AHB clock for the NVMCTRL_SMEEPROM is enabled. |

**Bit 21 – QSPI_2X** QSPI_2X AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the QSPI_2X is stopped. |
| 1 | The AHB clock for the QSPI_2X is enabled. |

**Bit 20 – PUKCC** PUKCC AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the PUKCC is stopped. |
| 1 | The AHB clock for the PUKCC is enabled. |

**Bit 19 – ICM** ICM AHB Clock Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The AHB clock for the ICM is stopped. |
| 1 | The AHB clock for the ICM is enabled. |

**Bits 17, 18 – CANn** CANn AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the CANn is stopped. |
| 1 | The AHB clock for the CANn is enabled. |

**Bits 15, 16 – SDHCn** SDHCn AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the SDHCn is stopped. |
| 1 | The AHB clock for the SDHCn is enabled. |

**Bit 14 – GMAC** GMAC AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the GMAC is stopped. |
| 1 | The AHB clock for the GMAC is enabled. |

**Bit 13 – QSPI** QSPI AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the QSPI is stopped. |
| 1 | The AHB clock for the QSPI is enabled. |

**Bit 12 – PAC** PAC AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the PAC is stopped. |
| 1 | The AHB clock for the PAC is enabled. |

**Bits 11,7,5 – Reserved** Reserved bits

Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to their reset value. If no reset value is given, write 0.

**Bit 10 – USB** USB AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the USB is stopped. |
| 1 | The AHB clock for the USB is enabled. |

**Bit 9 – DMAC** DMAC AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the DMAC is stopped. |
| 1 | The AHB clock for the DMAC is enabled. |

**Bit 8 – CMCC** CMCC AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the CMCC is stopped. |
| 1 | The AHB clock for the CMCC is enabled. |

**Bit 6 – NVMCTRL** NVMCTRL AHB Clock Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The AHB clock for the NVMCTRL is stopped. |
| 1 | The AHB clock for the NVMCTRL is enabled. |

**Bit 4 – DSU** DSU AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the DSU is stopped. |
| 1 | The AHB clock for the DSU is enabled. |

**Bit 3 – HPB3** AHB-APB Bridge D AHB Clock Enable AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the APBD is stopped. |
| 1 | The AHB clock for the APBD is enabled. |

**Bit 2 – HPB2** AHB-APB Bridge C AHB Clock Enable AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the APBC is stopped. |
| 1 | The AHB clock for the APBC is enabled. |

**Bit 1 – HPB1** AHB-APB Bridge B AHB Clock Enable AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the APBB is stopped. |
| 1 | The AHB clock for the APBB is enabled. |

**Bit 0 – HPB0** AHB-APB Bridge A AHB Clock Enable AHB Clock Enable

| Value | Description |
|---|---|
| 0 | The AHB clock for the APBA is stopped. |
| 1 | The AHB clock for the APBA is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 15.8.7. APBA Mask

**Name:** APBAMASK
**Offset:** 0x14
**Reset:** 0x000007FF
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TC1 | TC0 | SERCOM1 | SERCOM0 | FREQM | EIC | RTC | WDT |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GCLK | SUPC | OSC32KCTRL | OSCCTRL | RSTC | MCLK | PM | PAC |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 14, 15 – TCn** TCn APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the TCn is stopped. |
| 1 | The APBA clock for the TCn is enabled. |

**Bits 12, 13 – SERCOM** SERCOMn APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the SERCOMn is stopped. |
| 1 | The APBA clock for the SERCOMn is enabled. |

**Bit 11 – FREQM** FREQM APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the FREQM is stopped. |
| 1 | The APBA clock for the FREQM is enabled. |

**Bit 10 – EIC** EIC APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the EIC is stopped. |
| 1 | The APBA clock for the EIC is enabled. |

**Bit 9 – RTC** RTC APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the RTC is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The APBA clock for the RTC is enabled. |

**Bit 8 – WDT** WDT APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the WDT is stopped. |
| 1 | The APBA clock for the WDT is enabled. |

**Bit 7 – GCLK** GCLK APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the GCLK is stopped. |
| 1 | The APBA clock for the GCLK is enabled. |

**Bit 6 – SUPC** SUPC APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the SUPC is stopped. |
| 1 | The APBA clock for the SUPC is enabled. |

**Bit 5 – OSC32KCTRL** OSC32KCTRL APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the OSC32KCTRL is stopped. |
| 1 | The APBA clock for the OSC32KCTRL is enabled. |

**Bit 4 – OSCCTRL** OSCCTRL APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the OSCCTRL is stopped. |
| 1 | The APBA clock for the OSCCTRL is enabled. |

**Bit 3 – RSTC** RSTC APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the RSTC is stopped. |
| 1 | The APBA clock for the RSTC is enabled. |

**Bit 2 – MCLK** MCLK APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the MCLK is stopped. |
| 1 | The APBA clock for the MCLK is enabled. |

**Bit 1 – PM** PM APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the PM is stopped. |
| 1 | The APBA clock for the PM is enabled. |

**Bit 0 – PAC** PAC APBA Clock Enable

| Value | Description |
|---|---|
| 0 | The APBA clock for the PAC is stopped. |
| 1 | The APBA clock for the PAC is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 15.8.8. APBB Mask

**Name:** APBBMASK
**Offset:** 0x18
**Reset:** 0x00018056
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | RAMECC |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | TC3 | TC2 | TCC1 | TCC0 | SERCOM3 | SERCOM2 | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EVSYS | | | PORT | | NVMCTRL | DSU | USB |
| Access | R/W | | | R/W | | R/W | R/W | R/W |
| Reset | 0 | | | 1 | | 1 | 1 | 0 |

**Bit 16 – RAMECC** RAMECC APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the RAMECC is stopped. |
| 1 | The APBB clock for the RAMECC is enabled. |

**Bits 13, 14 – TCn** TCn APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the TCn is stopped. |
| 1 | The APBB clock for the TCn is enabled. |

**Bits 11, 12 – TCCn** TCCn APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the TCCn is stopped. |
| 1 | The APBB clock for the TCCn is enabled. |

**Bits 9, 10 – SERCOMn** SERCOMn APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the SERCOMn is stopped. |
| 1 | The APBB clock for the SERCOMn is enabled. |

**Bit 7 – EVSYS** EVSYS APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the EVSYS is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The APBB clock for the EVSYS is enabled. |

**Bit 4 – PORT** PORT APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the PORT is stopped. |
| 1 | The APBB clock for the PORT is enabled. |

**Bit 2 – NVMCTRL** NVMCTRL APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the NVMCTRL is stopped. |
| 1 | The APBB clock for the NVMCTRL is enabled. |

**Bit 1 – DSU** DSU APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the DSU is stopped. |
| 1 | The APBB clock for the DSU is enabled. |

**Bit 0 – USB** USB APBB Clock Enable

| Value | Description |
|---|---|
| 0 | The APBB clock for the USB is stopped. |
| 1 | The APBB clock for the USB is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 15.8.9. APBC Mask

**Name:** APBCMASK
**Offset:** 0x1C
**Reset:** 0x00002000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | CCL | QSPI | | ICM | TRNG | AES | AC |
| Access | | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | | 0 | 1 | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PDEC | TC5 | TC4 | TCC3 | TCC2 | GMAC | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bit 14 – CCL** CCL APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the CCL is stopped. |
| 1 | The APBC clock for the CCL is enabled. |

**Bit 13 – QSPI** QSPI APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the QSPI is stopped. |
| 1 | The APBC clock for the QSPI is enabled. |

**Bit 11 – ICM** ICM APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the ICM is stopped. |
| 1 | The APBC clock for the ICM is enabled. |

**Bit 10 – TRNG** TRNG APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the TRNG is stopped. |
| 1 | The APBC clock for the TRNG is enabled. |

**Bit 9 – AES** AES APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the AES is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The APBC clock for the AES is enabled. |

**Bit 8 – AC** AC APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the AC is stopped. |
| 1 | The APBC clock for the AC is enabled. |

**Bit 7 – PDEC** PDEC APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the PDEC is stopped. |
| 1 | The APBC clock for the PDEC is enabled. |

**Bits 5, 6 – TCn** TCn APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the TCn is stopped. |
| 1 | The APBC clock for the TCn is enabled. |

**Bits 3, 4 – TCCn** TCCn APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the TCCn is stopped. |
| 1 | The APBC clock for the TCCn is enabled. |

**Bit 2 – GMAC** GMAC APBC Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBC clock for the GMAC is stopped. |
| 1 | The APBC clock for the GMAC is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 15.8.10. APBD Mask

**Name:** APBDMASK
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PCC | I2S | DAC | ADC1 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADC0 | TC7 | TC6 | TCC4 | SERCOM7 | SERCOM6 | SERCOM5 | SERCOM4 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 11 – PCC** PCC APBD Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBD clock for the PCC is stopped. |
| 1 | The APBD clock for the PCC is enabled. |

**Bit 10 – I2S** I2S APBD Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBD clock for the I2S is stopped. |
| 1 | The APBD clock for the I2S is enabled. |

**Bit 9 – DAC** DAC APBD Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBD clock for the DAC is stopped. |
| 1 | The APBD clock for the DAC is enabled. |

**Bits 7, 8 – ADCn** ADCn APBD Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBD clock for the ADCn is stopped. |
| 1 | The APBD clock for the ADCn is enabled. |

**Bits 5, 6 – TCn** TCn APBD Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBD clock for the TCn is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The APBD clock for the TCn is enabled. |

**Bit 4 – TCC4** TCC4 APBD Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBD clock for the TCC4 is stopped. |
| 1 | The APBD clock for the TCC4 is enabled. |

**Bits 0, 1, 2, 3 – SERCOM** SERCOMn APBD Mask Clock Enable

| Value | Description |
|---|---|
| 0 | The APBD clock for the SERCOMn is stopped. |
| 1 | The APBD clock for the SERCOMn is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 16. RSTC – Reset Controller

## 16.1. Overview

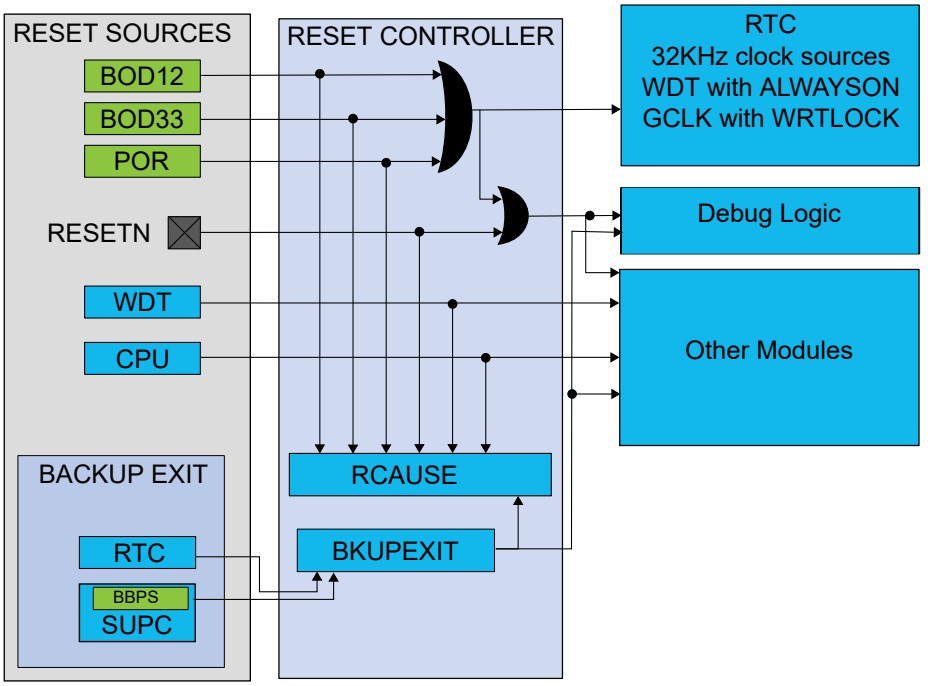
The Reset Controller (RSTC) manages the reset of the microcontroller. It issues a microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

## 16.2. Features

- Reset the microcontroller and set it to an initial state according to the reset source
- Reset cause register for reading the reset source from the application code
- Multiple reset sources
  - Power supply reset sources: POR, BOD12, BOD33
  - User reset sources: External reset (RESETN pin), Watchdog reset, and System Reset Request
  - Backup exit sources: Real-Time Counter (RTC) and Battery Backup Power Switch (BBPS)

## 16.3. Block Diagram

**Figure 16-1.** Reset System




## 16.4. Signal Description

| Signal Name | Type | Description |
|---|---|---|
| RESETN | Digital input | External reset pin |

One signal can be mapped on several pins.

**Related Links**

I/O Multiplexing and Considerations

## 16.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 16.5.1. I/O Lines

Not applicable.

### 16.5.2. Power Management

The Reset Controller module is always on.

### 16.5.3. Clocks

The RSTC bus clock (CLK_RSTC_APB) can be enabled and disabled in the Main Clock Controller.

**Related Links**
MCLK – Main Clock
Peripheral Clock Masking

### 16.5.4. DMA

Not applicable.

### 16.5.5. Interrupts

Not applicable.

### 16.5.6. Events

Not applicable.

### 16.5.7. Debug Operation

When the CPU is halted in debug mode, the RSTC continues normal operation.

### 16.5.8. Register Access Protection

All registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC).

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

### 16.5.9. Analog Connections

Not applicable.

## 16.6. Functional Description

### 16.6.1. Principle of Operation

The Reset Controller collects the various Reset sources and generates Reset for the device.

### 16.6.2. Basic Operation

#### 16.6.2.1. Initialization

After a power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the POR source.

#### 16.6.2.2. Enabling, Disabling, and Resetting

The RSTC module is always enabled.

#### 16.6.2.3. Reset Causes and Effects

The latest Reset cause is available in RCAUSE register, and can be read during the application boot sequence that followed the reset in order to determine proper action.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

These are the groups of Reset sources:

- Power supply Reset: Resets caused by an electrical issue. It covers POR and BODs Resets
- User Reset: Resets caused by the application. It covers external Resets, system Reset requests and watchdog Resets
- Backup reset: Resets caused by a Backup Mode exit condition

The following table lists the parts of the device that are reset, depending on the Reset type.

**Table 16-1.** Effects of the Different Reset Causes

| | Power Supply Reset | User Reset | | Backup Reset |
|---|---|---|---|---|
| | POR, BOD33, BOD12 | External Reset | WDT Reset, System Reset Request, NVM Reset | RTC, BBPS |
| RTC, OSC32KCTRL, RSTC | Reset | - | - | - |
| GCLK with WRTLOCK | Reset | - | - | Reset |
| Debug logic | Reset | Reset | - | Reset |
| Others | Reset | Reset | Reset | Reset |

The external Reset is generated when pulling the RESETN pin low.

The POR, BOD12, and BOD33 Reset sources are generated by their corresponding module in the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer.

The System Reset Request is a Reset generated by the CPU when asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details refer to the ARM® Cortex™ Technical Reference Manual on http://www.arm.com).

The NVM Reset is a Reset generated by the NVMCTRL when for example a BKSWRST command is performed (for details refer to NVMCTRL chapter).

From Backup Mode, the chip can be waken-up upon these conditions:

- Battery Backup Power Switch (BBPS): generated by the SUPC controller when the 3.3V VDDIO is restored.
- Real-Time Counter interrupt. For details refer to the applicable INTFLAG in the RTC for details.

If one of these conditions is triggered in Backup Mode, the RCAUSE.BACKUP bit is set and the Backup Exit Register (BKUPEXIT) is updated.

**Note:** Refer to the Timing Characteristics section of the Electrical Characteristics chapter.

**Related Links**
WDT – Watchdog Timer
SUPC – Supply Controller
Battery Backup Power Switch

### 16.6.3. Additional Features
Not applicable.

### 16.6.4. DMA Operation
Not applicable.

### 16.6.5. Interrupts
Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 16.6.6. Events

Not applicable.

### 16.6.7. Sleep Mode Operation

The RSTC module is active in all sleep modes.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

## 16.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | RCAUSE | 7:0 | BACKUP | SYST | WDT | EXT | NVM | BOD33 | BOD12 | POR |
| 0x01 | Reserved | | | | | | | | | |
| 0x02 | BKUPEXIT | 7:0 | HIB | | | | | BBPS | RTC | |

## 16.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 16.8.1. Reset Cause

**Name:** RCAUSE
**Offset:** 0x00
**Property:** –

When a Reset occurs, the bit corresponding to the Reset source is set to '1' and all other bits are written to '0'.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BACKUP | SYST | WDT | EXT | NVM | BOD33 | BOD12 | POR |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

**Bit 7 – BACKUP** Backup Reset
This bit is set if either a Backup or Hibernate Reset has occurred. Refer to BKUPEXIT register to identify the source of the Backup Reset.

**Bit 6 – SYST** System Reset Request
This bit is set if a System Reset Request has occurred. Refer to the Cortex processor documentation for more details.

**Bit 5 – WDT** Watchdog Reset
This bit is set if a Watchdog Timer Reset has occurred.

**Bit 4 – EXT** External Reset
This bit is set if an external Reset has occurred.

**Bit 3 – NVM** NVM Reset
This bit is set if an NVM Reset has occurred.

**Bit 2 – BOD33** Brown Out 33 Detector Reset
This bit is set if a BOD33 Reset has occurred.

**Bit 1 – BOD12** Brown Out 12 Detector Reset
This bit is set if a BOD12 Reset has occurred.

**Bit 0 – POR** Power On Reset
This bit is set if a POR has occurred.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 16.8.2. Backup Exit Source

**Name:** BKUPEXIT
**Offset:** 0x02
**Property:** –

When either a Hibernate ora Backup Reset occurs, the bit corresponding to the exit condition is set to '1', the other bits are written to '0'.

In some specific cases, the RTC and BBPS bits can be set together, e.g. when the device leaves the battery Backup Mode caused by a BBPS condition, and a RTC event was generated during the Battery Backup Mode period.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HIB | | | | | BBPS | RTC | |
| Access | R | | | | | R | R | |
| Reset | x | | | | | x | x | |

**Bit 7 – HIB** Hibernate
This bit is set if an Hibernate reset occurs. This bit is zero if a backup reset occurs.

**Bit 2 – BBPS** Battery Backup Power Switch
This bit is set if the Battery Backup Power Switch of the Supply Controller changes back from battery mode to main power mode.

**Bit 1 – RTC** Real Timer Counter Interrupt
This bit is set if an RTC interrupt flag is set in Backup Mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 17. RAMECC – RAM Error Correction Code (ECC)

## 17.1. Overview

For safety applications, the SAM D5x/E5x family can embed error correction codes (ECC) to detect and correct single bit errors or to enable dual error detection in SRAM. As discussed in Memories chapter, when the RAMECC is enabled, the top half of SRAM memory will be reserved to store error correction codes and will not be available for the application.

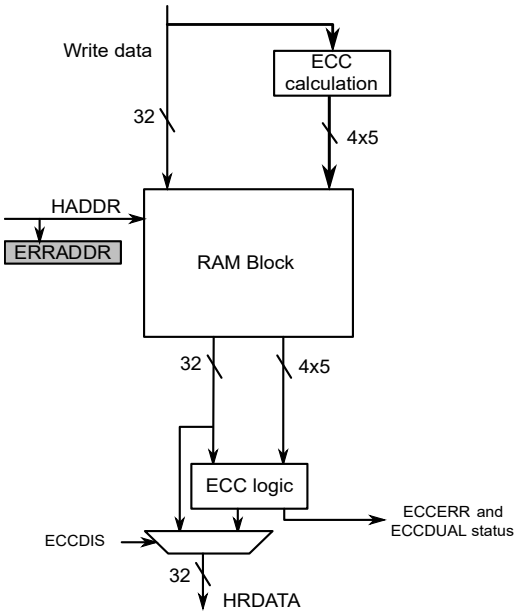
ECC calculation is software selectable through the RAM ECCDIS bit in the NVM User Row. For additional information, refer to Table 9-2.

## 17.2. Features

- Single bit correction and dual bit detection.
- Error Interrupt.

## 17.3. Block Diagram

**Figure 17-1.** RAMECC Block Diagram




## 17.4. Signal Description

Not applicable.

## 17.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 17.5.1. I/O Lines

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 17.5.2. Power Management

The RAMECC will continue to operate in any sleep mode where the selected source clock is running. The RAMECC's interrupts can be used to wake up the device from sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

**Related Links**

PM – Power Manager

### 17.5.3. Clocks

The RAMECC bus clock is provided by the Main Clock Controller (MCLK) through the AHB-APB B bridge. The clock is enabled and disabled by writing RAMECC bit the in the APB B Mask register (MCLK.APBBMASK.RAMECC). See the register description for the default state of the RAMECC bus clock.

**Related Links**

Peripheral Clock Masking

### 17.5.4. DMA

Not applicable.

### 17.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the RAMECC interrupt(s) requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 17.5.6. Events

Not applicable.

**Related Links**

Event System (EVSYS)

### 17.5.7. Debug Operation

When the CPU is halted in debug mode the RAMECC will correct and log ECC errors based on the table below.

**Table 17-1.** ECC Debug Operation

| DBGCTRL.ECCELOG | DBGCTRL.ECCDIS | Description |
|---|---|---|
| 0 | 0 | ECC errors from debugger reads are corrected but not logged in INTFLAG. |
| 1 | 0 | ECC errors from debugger reads are corrected and logged in INTFLAG. |
| X | 1 | ECC errors from debugger reads are not corrected or logged in INTFLAG. |

If the RAMECC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

### 17.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Status (STATUS) register.

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the Peripheral Access Controller chapter for details.

### 17.5.9. Analog Connections

Not applicable.

## 17.6. Functional Description

### 17.6.1. Principle of Operation

Error Correcting Code (ECC) is implemented to detect and correct errors that may arise in the RAM arrays. The ECC logic is capable of double error detection and single error correction per 8-bit byte.

Upon single bit error detection, the Single Bit Error interrupt flag is raised (INTFLAG.SINGLEE). If a dual error is detected, the Dual Error interrupt flag (INTFLAG.DUALE) is raised. When the first error is detected, the ERRADDR register is frozen with the failing address and remains frozen until INTFLAG.DUALE and INTFLAG.SINGLEE are cleared. If a dual bit error occurs while INTFLAG.SINGLEE is set, the ERRADDR register is updated with the dual bit error information and INTFLAG.DUALE is also set.

The INTFLAG.SINGLEE and INTFLAG.DUALE bits are both cleared on ERRADDR read.

The Figure 17-1 shows the ECC interface. When the ECC is disabled through its fuse bit, the ECC field in RAM is left unchanged on writes. On reads, ECC errors are not corrected or flagged.

**Related Links**

Block Diagram

### 17.6.2. Interrupts

The RAMECC has the following interrupt sources:

- Dual Bit Error (DUALE): Indicates that a dual bit error has been detected.
- Single Bit Error (SINGLEE): Indicates that a single bit error has been detected.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the ERRADDR register is read, the interrupt is disabled, or the RAMECC is reset.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

**Related Links**

Nested Vector Interrupt Controller

INTFLAG

Interrupt Flag Status and Clear


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 17.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | INTENCLR | 7:0 | | | | | | | DUALE | SINGLEE |
| 0x01 | INTENSET | 7:0 | | | | | | | DUALE | SINGLEE |
| 0x02 | INTFLAG | 7:0 | | | | | | | DUALE | SINGLEE |
| 0x03 | STATUS | 7:0 | | | | | | | | ECCDIS |
| 0x04 | ERRADDR | 7:0 | ERRADDR[7:0] | | | | | | | |
| | | 15:8 | ERRADDR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | ERRADDR[16] |
| | | 31:24 | | | | | | | | |
| 0x08 ... 0x0E | Reserved | | | | | | | | | |
| 0x0F | DBGCTRL | 7:0 | | | | | | | ECCELOG | ECCDIS |

## 17.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 17.8.1. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DUALE | SINGLEE |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – DUALE** Dual Bit Error Interrupt Enable Clear
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Dual Bit Error Interrupt Enable bit, which disables the Dual Bit Error interrupt.

| Value | Description |
|---|---|
| 0 | The Dual Bit Error interrupt is disabled. |
| 1 | The Dual Bit Error interrupt is enabled. |

**Bit 0 – SINGLEE** Single Bit Error Interrupt Enable Clear
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Single Bit Error Interrupt Enable bit, which disables the Single Bit Error interrupt.

| Value | Description |
|---|---|
| 0 | The Single Bit Error interrupt is disabled. |
| 1 | The Single Bit Error interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 17.8.2. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x01
**Reset:** 0x00
**Property:** Write-Protected

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DUALE | SINGLEE |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – DUALE** Dual Bit Error Interrupt Enable Set
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Dual Bit Error Interrupt Enable bit, which enables the Dual Bit Error interrupt.

| Value | Description |
|---|---|
| 0 | The Dual Bit Error interrupt is disabled. |
| 1 | The Dual Bit Error interrupt is enabled. |

**Bit 0 – SINGLEE** Single Bit Error Interrupt Enable Set
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Single Bit Error Interrupt Enable bit, which disables the Single Bit Error interrupt.

| Value | Description |
|---|---|
| 0 | The Single Bit Error interrupt is disabled. |
| 1 | The Single Bit Error interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 17.8.3. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x02
**Reset:** 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DUALE | SINGLEE |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – DUALE** Dual Bit ECC Error Interrupt
This flag is set on the occurrence of a dual bit ECC error.
Writing a '0' to this bit has no effect.
Reading the ECCADDR register will clear the Dual Bit Error interrupt flag.

| Value | Description |
|---|---|
| 0 | No dual bit errors have been received since the last clear. |
| 1 | At least one dual bit error has occurred since the last clear. |

**Bit 0 – SINGLEE** Single Bit ECC Error Interrupt
This flag is set on the occurrence of a single bit ECC error.
Writing a '0' to this bit has no effect.
Reading the ECCADDR register will clear the Single Bit Error interrupt flag.

| Value | Description |
|---|---|
| 0 | No errors have been received since the last clear. |
| 1 | At least one single bit error has occurred since the last clear. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 17.8.4. Status

**Name:** STATUS
**Offset:** 0x03
**Reset:** 0x00
**Property:** Read-only

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | ECCDIS |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

**Bit 0 – ECCDIS** ECC Disable
This bit is fuse updated at startup. When enabled, the calculated ECC is written to RAM along with data. ECC correction and detection is enabled for reads.

| Value | Description |
|---|---|
| 0 | ECC detection and correction is enabled. |
| 1 | ECC detection and correction is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 17.8.5. Error Address

**Name:** ERRADDR
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | ERRADDR[16] |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ERRADDR[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERRADDR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 16:0 – ERRADDR[16:0]** ECC Error Address

The RAM address offset from RAM start that caused an ECC error. If a single bit error is followed by a dual bit error, this register will be updated with the address of the dual bit error, otherwise it stalls on the first error occurrence. This register will read as zero unless INTFLAG.SINGLEE and/or INTFLAG.DUALE are 1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 17.8.6. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0F
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ECCELOG | ECCDIS |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ECCELOG** ECC Error Log
When DBGCTRL.ECCDIS=0, This bit controls whether ECC errors are logged in the INTFLAG register.
When DBGCTRL.ECCDIS=1, this bit has no meaning.

| Value | Description |
|---|---|
| 0 | ECC errors for debugger reads are not logged. |
| 1 | ECC errors for debugger reads are logged if DBGCTRL.ECCDIS=0. |

**Bit 0 – ECCDIS** ECC Disable
By default, ECC errors during debugger reads are corrected and logged based on DBGCTRL.ECCELOG. Setting this bit will disable ECC correction and logging.

| Value | Description |
|---|---|
| 0 | ECC errors are are corrected for debugger reads and logged based on DBGCTRL.ECCELOG. |
| 1 | ECC errors are masked for debugger reads. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 18. PM – Power Manager

**Related Links**

Sleep Mode Operation

## 18.1. Overview

The Power Manager (PM) controls the sleep modes and the power domain gating of the device.

Various sleep modes are provided in order to fit power consumption requirements. This enables the PM to stop unused modules in order to save power. In active mode, the CPU is executing application code. When the device enters a sleep mode, program execution is stopped and some modules and clock domains are automatically switched off by the PM according to the sleep mode. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the device from a sleep mode to active mode.

The user manually controls which power domains will be turned on and off in standby, hibernate and backup sleep mode.

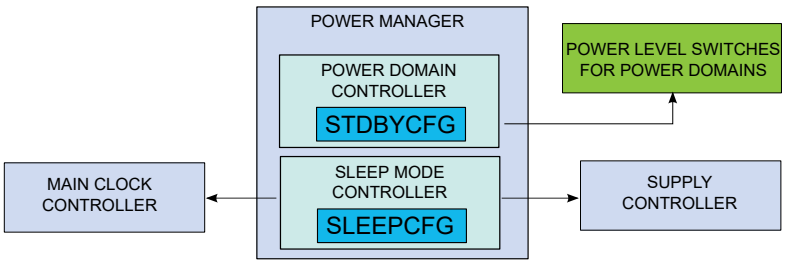
In backup and hibernate mode, the PM allows retaining the state of the I/O lines, preventing I/O lines from toggling during wake-up.

## 18.2. Features

- Power management control
  - Sleep modes: Idle, Hibernate, Standby, Backup, and Off
  - SleepWalking available in standby mode.
  - I/O lines retention in Backup mode

## 18.3. Block Diagram

**Figure 18-1.** PM Block Diagram




## 18.4. Signal Description

Not applicable.

## 18.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 18.5.1. I/O Lines

Not applicable.

### 18.5.2. Clocks

The PM bus clock (CLK_PM_APB) can be enabled and disabled in the Main Clock module. If this clock is disabled, it can only be re-enabled by a system reset.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.5.3. DMA

Not applicable.

### 18.5.4. Interrupts

The interrupt request line is connected to the interrupt controller. Using the PM interrupt requires the interrupt controller to be configured first.

### 18.5.5. Events

Not applicable.

### 18.5.6. Debug Operation

When the CPU is halted in debug mode, the PM continues normal operation. If standby sleep mode is requested by the system while in debug mode, the power domains are not turned off. As a consequence, power measurements while in debug mode are not relevant.

If Hibernate or Backup sleep mode is requested by the system while in debug mode, the core domains are kept on, and the debug modules are kept running to allow the debugger to access internal registers. When exiting the hibernate or backup mode upon a reset condition, the core domains are reset except the debug logic, allowing users to keep using their current debug session.

If OFF sleep mode is requested while in debug mode, the core domains are reset.

Hot plugging in standby mode is supported.

Hot plugging in Hibernate or backup mode or OFF mode is not supported as the DSU module is not powered.

Cold plugging in Hibernate or backup or OFF mode is supported if the external reset duration is superior to the corresponding sleep mode wakeup time (See Electrical characteristic chapter).

*Backup wakeup time is less than 200us in typical case. This value can be higher if voltage scaling in SUPC is enabled. Refers to SUPC for details.*

### 18.5.7. Register Access Protection

Registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC).

PAC write protection is not available for the following registers:

- Interrupt Flag register (INTFLAG). Refer to INTFLAG for details

Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

### 18.5.8. Analog Connections

Not applicable.

## 18.6. Functional Description

### 18.6.1. Terminology

The following is a list of terms used to describe the Power Managemement features of this microcontroller.

#### 18.6.1.1. Power Domains

Leaving aside the supply domains, such as VDDANA and VDDIO, the device is split into these power domains: PDCORESW, PDBACKUP, PDSYSRAM and PDBKUPRAM.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

PDCORESW, PDSYSRAM and PDBKUPRAM are "switchable power domains". In Standby, Hibernate or Backup mode, these power domains can be turned OFF to save leakage consumption according to user configuration.

- PDCORESW: contains the CPU and all the peripherals, except those located in the backup power domain.
- PDBACKUP: contains the backup peripherals: OSC32KCTRL, SUPC, RSTC, RTC and the PM itself.
- PDSYSRAM: contains the system RAM. It can be partially or fully turned OFF in Standby or Hibernate mode according to user configuration.
- PDBKUPRAM: contains the backup RAM. It can be partially or fully turned OFF in Backup mode.

#### 18.6.1.2. Sleep Modes

The device can be set in a sleep mode. In sleep mode, the CPU is stopped and the peripherals are either active or idle, according to the sleep mode depth:

- Idle sleep mode: The CPU is stopped. Synchronous clocks are stopped except when requested. The logic is retained.
- Standby sleep mode: The CPU is stopped as well as the peripherals. The logic is retained, and power domain gating can be used to fully or partially turn off the PDSYSRAM power domain.
- Hibernate sleep mode: The PDCORESW power domain is turned off. The backup power domain is kept powered to allow few features to run (RTC, 32 kHz clock sources, and wake-up from RTC Tamper pins). The PDSYSRAM power domain can be retained according to software configuration.
- Backup sleep mode: Only the backup domain is kept powered to allow few features to run (RTC, 32 kHz clock sources, and wake-up from the RTC Tamper pins). The PDBKUPRAM power domain can be retained according to software configuration.
- Off sleep mode: The entire device is powered off.

### 18.6.2. Principle of Operation

In active mode, all clock domains and power domains are active, allowing software execution and peripheral operation. The PM Sleep Mode Controller allows to save power by choosing between different sleep modes depending on application requirements, see Sleep Mode Controller.

The PM Power Domain Controller allows to reduce the power consumption in standby mode even further.

### 18.6.3. Basic Operation

#### 18.6.3.1. Initialization

After a Power-on Reset (POR), the PM is enabled, the device is in Active mode.

#### 18.6.3.2. Enabling, Disabling and Resetting

The PM is always enabled and can not be reset.

#### 18.6.3.3. Sleep Mode Controller

A Sleep mode is entered by executing the Wait For Interrupt instruction (WFI). The Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE) select the level of the sleep mode.

**Note:** A small latency happens between the store instruction and actual writing of the SLEEPCFG.SLEEPCFG register due to bridges. Software must ensure that the SLEEPCFG register reads the desired value before executing a WFI instruction.

**Note:** After power-up, the MAINVREG low power mode takes some time to stabilize. Once stabilized, the INTFLAG.SLEEPRDY bit is set. Before entering Standby, Hibernate or Backup mode, software must ensure that the INTFLAG.SLEEPRDY bit is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 18-1.** Sleep Mode Entry and Exit Table

| Mode | Mode Entry | Wake-Up Sources |
|---|---|---|
| IDLE | SLEEPCFG.SLEEPMODE = IDLE | Synchronous (2) (APB, AHB), asynchronous (1) |
| STANDBY | SLEEPCFG.SLEEPMODE = STANDBY | Synchronous (3), asynchronous (1) |
| HIBERNATE | SLEEPCFG.SLEEPMODE = HIBERNATE | Hibernate reset detected by the RSTC |
| BACKUP | SLEEPCFG.SLEEPMODE = BACKUP | Backup reset detected by the RSTC |
| OFF | SLEEPCFG.SLEEPMODE = OFF | External Reset |

**Notes:**

1. Asynchronous: interrupt generated on generic clock, external clock, or external event.
2. Synchronous: interrupt generated on synchronous (APB or AHB) clock.
3. Synchronous interrupt only for peripherals configured to run in standby.

**Note:** The type of wake-up sources (synchronous or asynchronous) is given in each module interrupt section.

The sleep modes (idle, standby, hibernate, backup, and off) and their effect on the clocks activity, the regulator and the NVM state are described in the table and the sections below. Refer to Power Domain Controller for the power domain gating effect.

**Table 18-2.** Sleep Mode Overview

| Mode | Main clock | CPU | AHBx and APBx clock | GCLK clocks | Oscillators | | Regulator | NVM |
|---|---|---|---|---|---|---|---|---|
| | | | | | ONDEMAND = 0 | ONDEMAND = 1 | | |
| Active | Run | Run | Run | Run(1) | Run | Run if requested | MAINVREG | active |
| IDLE | Run | Stop | Stop(2) | Run(1) | Run | Run if requested | MAINVREG | active |
| STANDBY | Stop | Stop | Stop(2) | Stop(2) | Run if requested or RUNSTDBY=1 | Run if requested | MAINVREG in low power mode | Ultra Low power |
| HIBERNATE | Stop | Stop | Stop | Stop | Stop | Stop | MAINVREG in low power mode | Ultra Low power+ |
| BACKUP | Stop | Stop | Stop | Stop | Stop | Stop | Backup regulator (LPVREG) | OFF |
| OFF | Stop | Stop | Stop | OFF | OFF | OFF | OFF | OFF |

**Notes:**

1. Running if requested by peripheral during SleepWalking
2. Running during SleepWalking

#### 18.6.3.3.1. IDLE Mode

IDLE mode allows power optimization with the fastest wake-up time.

The CPU is stopped, and peripherals are still working. As in Active mode, the AHBx and APBx clocks for peripheral are still provided if requested. As the main clock source is still running, wake-up time is very fast.

- Entering Idle mode: The Idle mode is entered by executing the WFI instruction. Additionally, if the SLEEPONEXIT bit in the Cortex System Control register (SCR) is set, the Idle mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine, refer to the ARM Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Before entering the Idle mode, the user must select the Idle Sleep mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=IDLE).
- Exiting Idle mode: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry. The system goes back to the Active mode. The CPU and affected modules are restarted.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

GCLK clocks, regulators and RAM are not affected by the Idle Sleep mode and operate in normal mode.

#### 18.6.3.3.2. STANDBY Mode

The STANDBY mode is the lowest power configuration while keeping the state of the logic and the content of the RAM.

In this mode, all clocks are stopped except those configured to be running sleepwalking tasks. The clocks can also be active on request or at all times, depending on their on-demand and run-in-standby settings. Either synchronous (CLK_APBx or CLK_AHBx) or generic (GCLK_x) clocks or both can be involved in sleepwalking tasks. This is the case when for example the SERCOM RUNSTDBY bit is written to '1'.

- Entering STANDBY mode: This mode is entered by executing the WFI instruction after writing the Sleep Mode bit in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=STANDBY). The SLEEPONEXIT feature is also available as in IDLE mode.
- Exiting STANDBY mode: Any peripheral able to generate an asynchronous interrupt can wake up the system. For example, a peripheral running on a GCLK clock can trigger an interrupt. When the enabled asynchronous wake-up event occurs and the system is woken up, the device will either execute the interrupt service routine or continue the normal program execution according to the Priority Mask Register (PRIMASK) configuration of the CPU.

Refer to the section about the Power Domain Controller for the RAM state.

The regulator operates in low-power mode by default and switches automatically to the normal mode in case of a sleepwalking task requiring more power. It returns automatically to low power mode when the sleepwalking task is completed.

**Related Links**

Power Domain Controller

#### 18.6.3.3.3. Hibernate and Backup Mode

Hibernate and Backup mode allow achieving the lowest power consumption aside from OFF. The device is entirely powered off except for the backup domain. All peripherals in backup domain are allowed to run, for example, the RTC can be clocked by a 32.768 kHz oscillator. All PM registers are retained except INTENCLR, INTENSET, INTFLAG, and SLEEPCFG registers.

- Entering Hibernate or Backup mode: This mode is entered by executing the WFI instruction after selecting the Hibernate or Backup mode by writing the Sleep Mode bits in the Sleep Configuration register (SLEEPCFG.SLEEPMODE=HIBERNATE or =BACKUP).
- Exiting Hibernate or Backup mode: is triggered when a Hibernate or Backup Reset is detected by the Reset Controller (RSTC).
  **Note:** In Hibernate mode, the MAINVREG (in low-power mode) regulator is used to allow powering the PDRAM power domain which can be fully retained according to software configuration.

  **Note:** In Backup mode, the backup regulator (LPVREG) is used, unless VREG.RUNBKUP = 1. When VREG.RUNBKUP is set, the Main regulator is used in backup mode. The PDBKUPRAM power domain can be fully retained according to software configuration.
  Refer to the Power Domain Controller for the RAM state.

#### 18.6.3.3.4. OFF Mode

In Off mode, the device is entirely powered-off.

- Entering Off mode: This mode is entered by selecting the Off mode in the Sleep Configuration register by writing the Sleep Mode bits (SLEEPCFG.SLEEPMODE=OFF), and subsequent execution of the WFI instruction.
- Exiting Off mode: This mode is left by pulling the $\overline{\text{RESET}}$ pin low, or when a power Reset is done.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 18.6.3.4. I/O Lines Retention in HIBERNATE or BACKUP Mode

When entering HIBERNATE or BACKUP mode, the PORT is powered off but the pin configuration is retained. When the device exits the HIBERNATE or BACKUP mode, the I/O line configuration can either be released or stretched, based on the I/O Retention bit in the Control A register (CTRLA.IORET).

- If IORET=0 when exiting HIBERNATE or BACKUP mode, the I/O lines configuration is released and driven by the reset value of the PORT.
- If the IORET=1 when exiting HIBERNATE or BACKUP mode, the configuration of the I/O lines is retained until the IORET bit is written to 0. It allows the I/O lines to be retained until the application has programmed the PORT.

#### 18.6.3.5. Power Domain Controller

The Power Domain Controller provides several ways of how power domains are handled while the device is in standby, hibernate or backup mode:

- Standby mode:
  When entering standby mode, the PDSYSRAM power domain can be either fully or partially retained or be fully off according to STDBYCFG.RAMCFG bits. When running sleepwalking task, PDSYSRAM power domain is active whatever the STDBYCFG.RAMCFG bits are.
- Hibernate mode:
  When entering hibernate mode, the PDCORESW power domain is off. As in standby mode, the PDSYSRAM power domain can be selectively turned ON or OFF by using the HIBCFG.RAMCFG bits. PDBKUPRAM power domain can be either fully or partially retained or be fully off according to HIBCFG.BRAMCFG bits. If partial option is selected, only the lowest 4KBytes section is retained
- Backup mode:
  When entering backup mode, the PDCORESW and PDSYSRAM power domains are off. PDBACKUP is still active. As in hibernate mode, PDBKUPRAM power domain can be either fully or partially retained or be fully off according to BKUPCFG.BRAMCFG bits.
- OFF mode:
  When entering OFF mode, all the power domains are off.

The table below illustrates the PDRAM state:

**Table 18-3.** Sleep Mode versus PDSYSRAM Power Domain State Overview

| | | | Power Domain State | | |
|---|---|---|---|---|---|
| Sleep Mode | STDBYCFG.RAMCFG | HIBCFG.RAMCFG | PDCORESW | PDBACKUP | PDSYSRAM |
| Active | N/A | N/A | active | active | active |
| Idle | N/A | N/A | active | active | active |
| Standby with sleepwalking | N/A | N/A | active | active | active |
| Standby - case 1 | RET | N/A | active | active | retained |
| Standby - case 2 | PARTIAL | N/A | active | active | 32K retained |
| Standby - case 3 | OFF | N/A | active | active | off |
| Hibernate - case 1 | N/A | RET | off | active | retained |
| Hibernate - case 2 | N/A | PARTIAL | off | active | 32K retained |
| Hibernate - case 3 | N/A | OFF | off | active | off |
| Backup | N/A | N/A | off | active | off |
| Off | N/A | N/A | off | off | off |

The table below illustrates the PDBKUPRAM state:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 18-4.** Sleep Mode versus PDBKUPRAM Power Domain State Overview

| | | | Power Domain State | | |
|---|---|---|---|---|---|
| Sleep Mode | HIBCFG.BRAMCFG | BKUPCFG.BRAMCFG | PDCORESW | PDBACKUP | PDBKUPRAM |
| Active | N/A | N/A | active | active | active |
| Idle | N/A | N/A | active | active | active |
| Standby | N/A | N/A | active | active | retained |
| Hibernate - case 1 | RET | N/A | off | active | retained |
| Hibernate - case 2 | PARTIAL | N/A | off | active | 4KB retained |
| Hibernate - case 3 | OFF | N/A | off | active | off |
| Backup | N/A | RET | off | active | retained |
| Backup | N/A | PARTIAL | off | active | 4KB retained |
| Backup | N/A | OFF | off | active | off |
| Off | N/A | N/A | off | off | off |

### 18.6.3.6. Regulators, RAMs, and NVM State in Sleep Mode

By default, in standby sleep mode and backup sleep mode, the RAMs, NVM, and regulators are automatically set in low-power mode in order to reduce power consumption:

- The RAM is in low-power mode if the device is in standby mode.
- Non-Volatile Memory - the NVM is automatically set in low power mode in these conditions:
  - When the device is in standby sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
  - When the device is in idle sleep mode and the NVM is not accessed. This behavior can be changed by software by configuring the SLEEPPRM bit group of the CTRLB register in the NVMCTRL peripheral.
- Regulators: by default, in standby sleep mode, the PM analyzes the device activity to use either the main or the low-power voltage regulator to supply the VDDCORE.

GCLK clocks, regulators and RAM are not affected in idle sleep mode and will operate as normal.

**Table 18-5.** Regulators, RAMs, and NVM state in Sleep Mode

| Sleep Mode | SRAM Mode | NVM | Regulators | | |
|---|---|---|---|---|---|
| | | | VDDCORE | | VDDBU |
| | | | main | ULP | |
| Active | normal | normal | on | on | on |
| Idle | auto(1) | on | on | on | on |
| Standby - case 1 | normal | auto(1) | auto(2) | on | on |
| Standby - case 2 | low power | low power | auto(2) | on | on |
| Standby - case 3 | low power | low power | auto(2) | on | on |
| Standby - case 4 | low power | low power | off | on | on |
| Backup | off | off | off | off | on |
| OFF | off | off | off | off | off |

**Notes:**

1. auto: by default, NVM is in low-power mode if not accessed.
2. auto: by default, the main voltage regulator is on if GCLK, APBx, or AHBx clock is running during SleepWalking.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Power Domain Controller

## 18.6.4. Advanced Features

### 18.6.4.1. SleepWalking

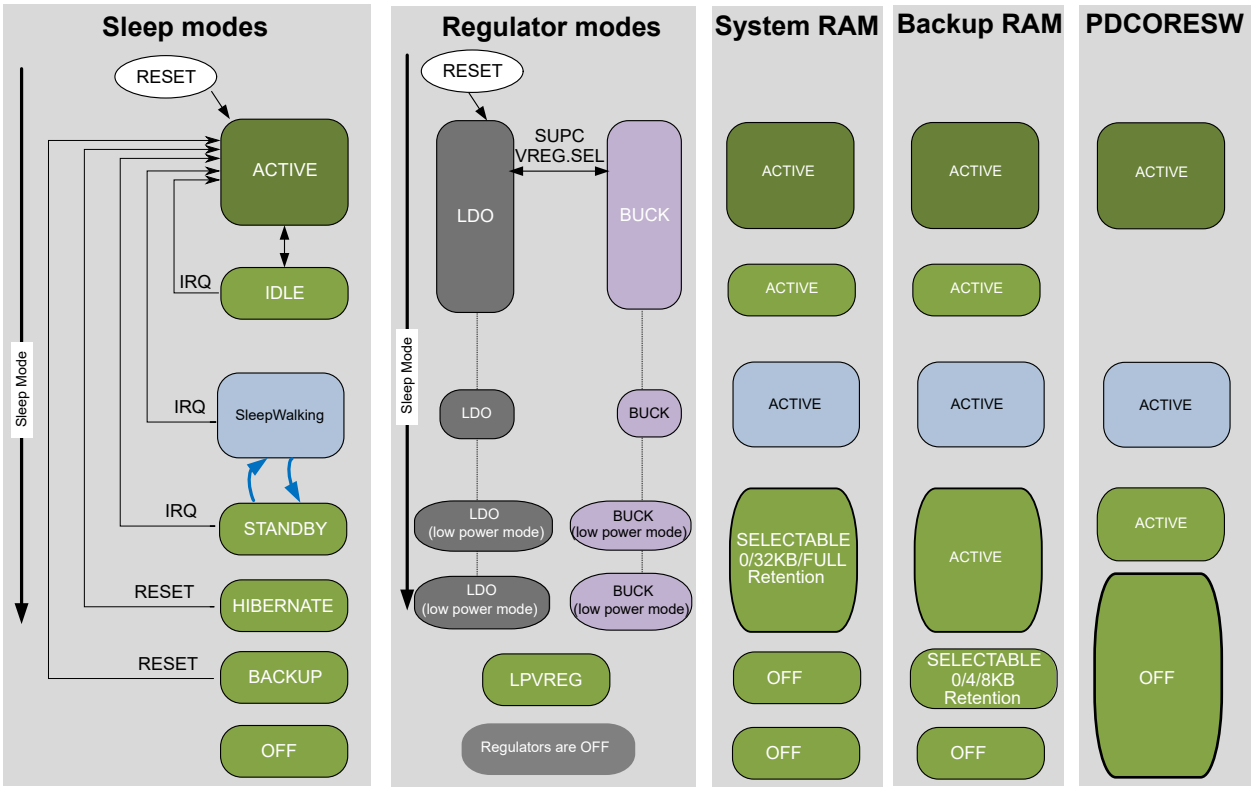
SleepWalking is the capability for a device to temporarily wake up clocks for a peripheral to perform a task without waking up the CPU from STANDBY sleep mode. At the end of the sleepwalking task, the device can either be woken p by an interrupt (from a peripheral involved in SleepWalking) or enter again into STANDBY sleep mode. In this device, SleepWalking is supported only on GCLK clocks by using the on-demand clock principle of the clock sources.

In standby, when SleepWalking is ongoing:

- All the power domains are turned ON including PDRAM power domain.
- The MAINVREG regulator used to execute the sleepwalking task is the selected regulator used in active mode (LDO or Buck converter). Low power mode of the MAINVREG is not activated during sleepwalking.

These are illustrated in the figure below.

**Figure 18-2.** Operating Conditions and SleepWalking




### 18.6.4.2. Wake-Up Time

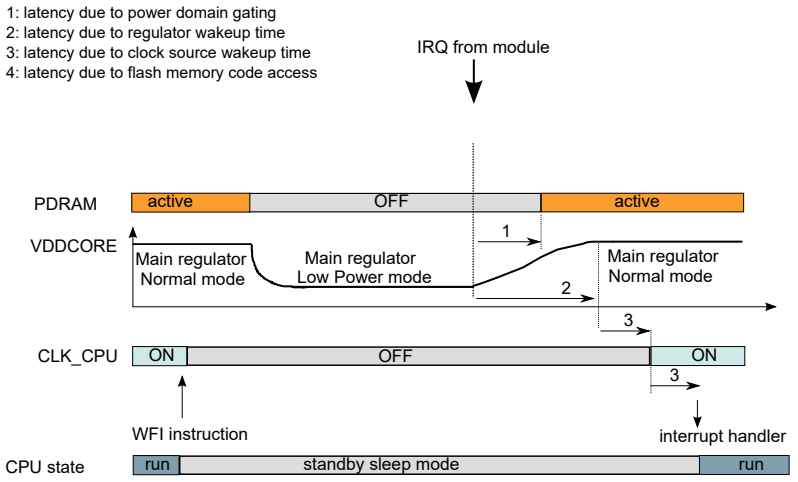
As shown in the figure below, total wake-up time depends on:

- Latency due to Power Domain Gating:
  Usually, wake-up time is measured with the assumption that the power domains are already in active state. When using Power Domain Gating, changing a power domain from OFF to active state will take a certain time, refer to Electrical Characteristics. If all power domains were already in active state in standby sleep mode, this latency is zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Latency due to Regulator effect:
  As example, if the device is in standby sleep mode using the main voltage regulator (MAINVREG) in low power mode, the voltage level is lower than the one used in active mode. When the device wakes up, it takes a certain amount of time for the main regulator to transition to the voltage level corresponding to active mode, causing additional wake-up time.
- Latency due to the CPU clock source wake-up time.
- Latency due to the NVM memory access.
  **Note:** NVM and MAINVREG latencies can be reduced by setting the Fast Wake-Up bits in the Standby Configuration register (STDBYCFG.FASTWKUP).

**Figure 18-3.** Total Wake-up Time from Standby Sleep Mode




**Related Links**
Power Domains

### 18.6.5. DMA Operation
Not applicable.

### 18.6.6. Interrupts
The peripheral has the following interrupt sources:

- Sleep Mode Entry Ready (SLEEPRDY): indicates that the device is ready to enter standby, hibernate or backup sleep mode.
  This interrupt is a synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the peripheral is reset.

An interrupt flag is cleared by writing a '1' to the corresponding bit in the INTFLAG register. Each peripheral can have one interrupt request line per interrupt source or one common interrupt request line for all the interrupt sources. If the peripheral has one common interrupt request line for all the interrupt sources, the user must read the INTFLAG register to determine which interrupt condition is present.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.6.7. Events

Not applicable.

### 18.6.8. Sleep Mode Operation

The Power Manager is always active.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

## 18.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | IORET | | |
| 0x01 | SLEEPCFG | 7:0 | | | | | | SLEEPMODE[2:0] | | |
| 0x02 ... 0x03 | Reserved | | | | | | | | | |
| 0x04 | INTENCLR | 7:0 | | | | | | | | SLEEPRDY |
| 0x05 | INTENSET | 7:0 | | | | | | | | SLEEPRDY |
| 0x06 | INTFLAG | 7:0 | | | | | | | | SLEEPRDY |
| 0x07 | Reserved | | | | | | | | | |
| 0x08 | STDBYCFG | 7:0 | | | FASTWKUP[1:0] | | | | RAMCFG[1:0] | |
| 0x09 | HIBCFG | 7:0 | | | | | BRAMCFG[1:0] | | RAMCFG[1:0] | |
| 0x0A | BKUPCFG | 7:0 | | | | | | | BRAMCFG[1:0] | |
| 0x0B ... 0x11 | Reserved | | | | | | | | | |
| 0x12 | PWSAKDLY | 7:0 | IGNACK | DLYVAL[6:0] | | | | | | |

## 18.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection section.

**Related Links**

Register Access Protection


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | IORET | | |
| Access | | | | | | R/W | | |
| Reset | | | | | | 0 | | |

**Bit 2 – IORET** I/O Retention

**Note:** This bit is not reset by a hibernate or backup reset. When the IORET feature is used, the debugger access to the chip will not be allowed until the IORET bit is cleared after waking up from hibernate or backup sleep. When the IORET is set in active mode, the PORT can still be controlled by peripherals and the PORT registers. It is only when the device wakes up from hibernate or backup sleep mode that the IORET= 1 will prevent the PORT from being controlled by the peripherals or PORT registers. POR and BOD33 resets can clear the IORET bit.

| Value | Description |
|---|---|
| 0 | After waking up from Hibernate or Backup mode, I/O lines are not held. |
| 1 | After waking up from Hibernate or Backup mode, I/O lines are held until IORET is written to 0. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.2. Sleep Configuration

**Name:** SLEEPCFG
**Offset:** 0x01
**Reset:** 0x02
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SLEEPMODE[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 2:0 – SLEEPMODE[2:0]** Sleep Mode
**Note:** A small latency happens between the store instruction and actual writing of the SLEEPCFG register due to bridges. Software has to make sure the SLEEPCFG register reads the wanted value before executing WFI instruction.

| Value | Name | Definition |
|---|---|---|
| 0x0 | Reserved | - |
| 0x1 | Reserved | - |
| 0x2 | IDLE | CPU, AHBx, and APBx clocks are OFF |
| 0x3 | Reserved | Reserved |
| 0x4 | STANDBY | All Clocks are OFF |
| 0x5 | HIBERNATE | Backup domain is ON as well as some PDRAMs |
| 0x6 | BACKUP | Only Backup domain is powered ON |
| 0x7 | OFF | All power domains are powered OFF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.3. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SLEEPRDY |
| Access | | | | | | | | W |
| Reset | | | | | | | | 0 |

**Bit 0 – SLEEPRDY** Sleep Mode Entry Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Sleep Mode Entry Ready Interrupt Enable bit and the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Sleep Mode Entry Ready interrupt is disabled. |
| 1 | The Sleep Mode Entry Ready interrupt is enabled and will generate an interrupt request when the Sleep Mode Entry Ready Interrupt Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.4. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SLEEPRDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – SLEEPRDY** Sleep Mode Entry Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Sleep Mode Entry Ready Interrupt Enable bit and enable the Sleep Mode Entry Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Sleep Mode Entry Ready interrupt is disabled. |
| 1 | The Sleep Mode Entry Ready interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.5. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x06
**Reset:** 0x00
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SLEEPRDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – SLEEPRDY** Sleep Mode Entry Ready
This flag is set when the main very low power mode is ready and will generate an interrupt if INTENCLR/SET.SLEEPRDY is '1'. See this Note for details.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Performance Ready interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 18.8.6. Hibernate Configuration

**Name:** HIBCFG
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | BRAMCFG[1:0] | | RAMCFG[1:0] | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 3:2 – BRAMCFG[1:0]** Backup RAM Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | RET | In hibernate mode, all the backup RAM is retained. |
| 0x1 | PARTIAL | In hibernate mode, only the first 4Kbytes of the backup RAM is retained. |
| 0x2 | OFF | In hibernate mode, all the backup RAM is turned OFF. |
| 0x3 | Reserved | Reserved. |

**Bits 1:0 – RAMCFG[1:0]** RAM Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | RET | In hibernate mode, all the system RAM is retained. |
| 0x1 | PARTIAL | In hibernate mode, only the first 32Kbytes of the system RAM is retained. |
| 0x2 | OFF | In hibernate mode, all the system RAM is turned OFF. |
| 0x3 | Reserved | Reserved. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.7. Standby Configuration

**Name:** STDBYCFG
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | FASTWKUP[1:0] | | | | RAMCFG[1:0] | |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bits 5:4 – FASTWKUP[1:0]** Fast Wakeup

| Value | Name | Description |
|---|---|---|
| 0x0 | NO | Fast Wakeup is disabled. |
| 0x1 | NVM | Fast Wakeup is enabled on NVM. |
| 0x2 | MAINVREG | Fast Wakeup is enabled on the main voltage regulator (MAINVREG). |
| 0x3 | BOTH | Fast Wakeup is enabled on both NVM and MAINVREG.. |

**Bits 1:0 – RAMCFG[1:0]** RAM Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | RET | In standby mode, all the system RAM is retained. |
| 0x1 | PARTIAL | In standby mode, only the first 32Kbytes of the system RAM is retained. |
| 0x2 | OFF | In standby mode, all the system RAM is turned OFF. |
| 0x3 | Reserved | Reserved. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.8. Backup Configuration

**Name:** BKUPCFG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BRAMCFG[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – BRAMCFG[1:0]** Backup RAM Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | RET | In backup mode, all the backup RAM is retained. |
| 0x1 | PARTIAL | In backup mode, only the first 4Kbytes of the backup RAM is retained. |
| 0x2 | OFF | In backup mode, all the backup RAM is turned OFF. |
| 0x3 | Reserved | Reserved. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 18.8.9. Global Status

**Name:** PWSAKDLY
**Offset:** 0x12
**Reset:** 0x00
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IGNACK | DLYVAL[6:0] | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – IGNACK** Ignore Acknowledge signal

| Value | Description |
|---|---|
| 0 | Power Switch acknowledge signal is taken into account when entering/exiting retention mode. According to the DLYVAL field, a supplementary delay is also added (from 0 to 127 digital ring oscillator period). |
| 1 | Power Switch acknowledge signal is ignored when entering/exiting retention mode, and is replaced by a overflow counter signal clocked on internal digital ring oscillator. The overflow counter is programmable by using the DLYVAL field. |

**Bits 6:0 – DLYVAL[6:0]** Delay value
Value of the counter overflow. See the IGNACK bit description to get more details.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 19. SUPC – Supply Controller

## 19.1. Overview

The Supply Controller (SUPC) manages the voltage reference, power supply, and supply monitoring of the device. It is also able to control two output pins.

The SUPC controls the voltage regulators for the core (VDDCORE) and backup (VDDBU) domains. It sets the voltage regulators according to the sleep modes, or the user configuration. In active mode, the voltage regulators can be selected on-the-fly between LDO (low-dropout) type regulator or Buck converter.

The SUPC supports connection of a battery backup to the VBAT power pin. It includes functionality that enables automatic power switching between main power and battery backup power. This ensures power to the backup domain when the main battery or power source is unavailable.

The SUPC embeds two Brown-Out Detectors: BOD33 monitors the voltage applied to the device (VDD or VBAT) and BOD12 monitors the internal voltage to the core (VDDCORE). The BOD33 can monitor the supply voltage continuously (continuous mode) or periodically (sampling mode), in normal or low power mode.

The SUPC generates also a selectable reference voltage and a voltage dependent on the temperature which can be used by analog modules such as the ADC and DAC.

## 19.2. Features

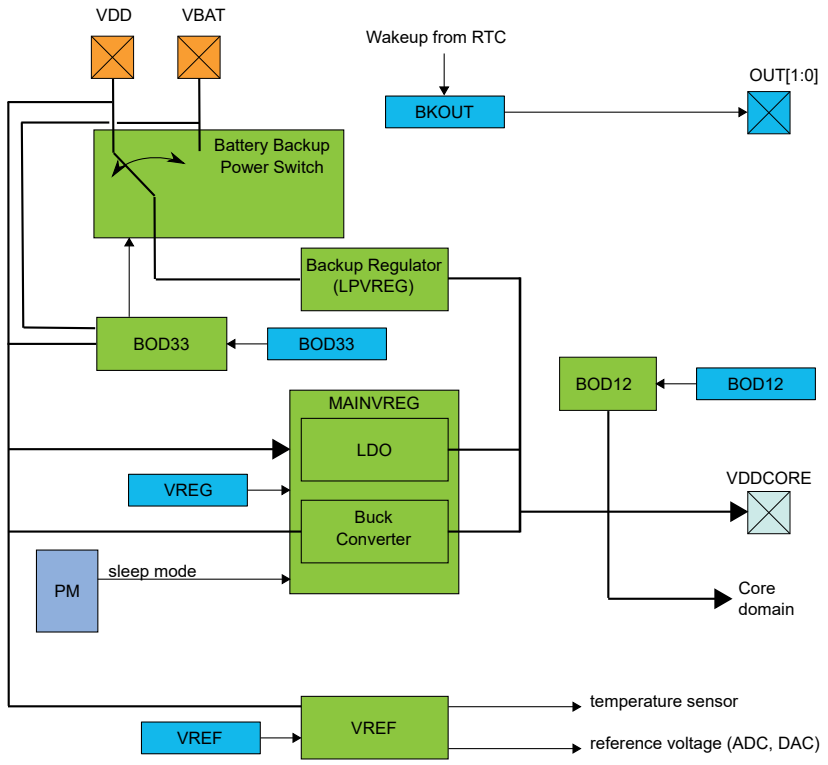
- Voltage Regulator System
  - Main voltage regulator: LDO or Buck Converter in Active, Standby or Hibernate mode (MAINVREG)
  - Low-Power voltage regulator in Backup mode (LPVREG)
  - Controlled VDDCORE voltage slope when changing VDDCORE
- Battery Backup Power Switch
  - Automatic switching from main power to battery backup power
    - Automatic entry to backup mode when switched to battery backup power
  - Automatic switching from battery backup power to main power
    - Automatic exit from backup mode when switched back to main power
    - Stay in backup mode when switched back to main power
- Voltage Reference System
  - Reference voltage for ADC and DAC
  - Temperature sensor
- 3.3V Brown-Out Detector (BOD33)
  - Programmable threshold
  - Threshold value loaded from NVM User Row at startup
  - Triggers resets, interrupts, or Battery Backup Power Switch. Action loaded from NVM User Row
  - Operating modes:
    - Continuous mode
    - Low power and Sampled mode for low power applications with programmable sample frequency
  - Hysteresis value from Flash User Calibration


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - Monitor VDD or VBAT
- 1.2V Brown-Out Detector (BOD12)
- Output pins
  - Pin toggling on RTC event

## 19.3. Block Diagram

**Figure 19-1.** SUPC Block Diagram




## 19.4. Signal Description

| Signal Name | Type | Description |
|---|---|---|
| OUT[1:0] | Digital Output | SUPC Outputs |

One signal can be mapped on several pins.

**Related Links**

I/O Multiplexing and Considerations

## 19.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 19.5.1. I/O Lines

I/O lines are configured by SUPC when the SUPC output (signal OUT) is enabled. The I/O lines need no user configuration.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 19.5.2. Power Management

The SUPC can operate in all sleep modes except backup sleep mode. BOD33 and Battery backup Power Switch can operate in backup mode.

**Related Links**
PM – Power Manager

### 19.5.3. Clocks

The SUPC bus clock (CLK_SUPC_APB) can be enabled and disabled in the Main Clock module.

A 32KHz clock, asynchronous to the user interface clock (CLK_SUPC_APB), is required to run BOD33 and in sampled mode. Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**
OSC32KCTRL – 32KHz Oscillators Controller
Peripheral Clock Masking

### 19.5.4. DMA

Not applicable.

### 19.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the SUPC interrupts requires the interrupt controller to be configured first.

**Related Links**
Nested Vector Interrupt Controller

### 19.5.6. Events

Not applicable.

### 19.5.7. Debug Operation

When the CPU is halted in debug mode, the SUPC continues normal operation. If the SUPC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

If a cold plug-in is detected by the system, BOD33 and BOD12 will use the factory calibration setting instead of the user calibration. In hot plug-in, the BODs resets keep running.

### 19.5.8. Register Access Protection

Registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC).

**Note:** Not all registers with write access can be write-protected.

PAC write protection is not available for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description.

**Related Links**
PAC - Peripheral Access Controller

### 19.5.9. Analog Connections

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 19.6. Functional Description

### 19.6.1. Voltage Regulator System Operation

#### 19.6.1.1. Enabling, Disabling, and Resetting

The LDO main voltage regulator is enabled after a power-reset. The main voltage regulator output supply level is automatically defined by the sleep mode selected in the Power Manager module (PM.SLEEPCFG.SLEEPMODE).

#### 19.6.1.2. Initialization

After a power-reset, the LDO voltage regulator supplying VDDCORE is enabled.

#### 19.6.1.3. Selecting a Voltage Regulator

In Active mode, the type of the main voltage regulator supplying VDDCORE can be switched on the fly. The two alternatives are a LDO regulator and a Buck converter.

The main voltage regulator switching sequences are as follows:

- The user changes the value of the Voltage Regulator Selection bit in the Voltage Regulator System Control register (VREG.SEL)
- The start of the switching sequence is indicated by clearing the Voltage Regulator Ready bit in the STATUS register (STATUS.VREGRDY=0)
- Once the switching sequence is completed, STATUS.VREGRDY will read '1'

The Voltage Regulator Ready (VREGRDY) interrupt can also be used to detect a zero-to-one transition of the STATUS.VREGRDY bit.

#### 19.6.1.4. Voltage Scaling Control

The VDDCORE supply will change under certain circumstances:

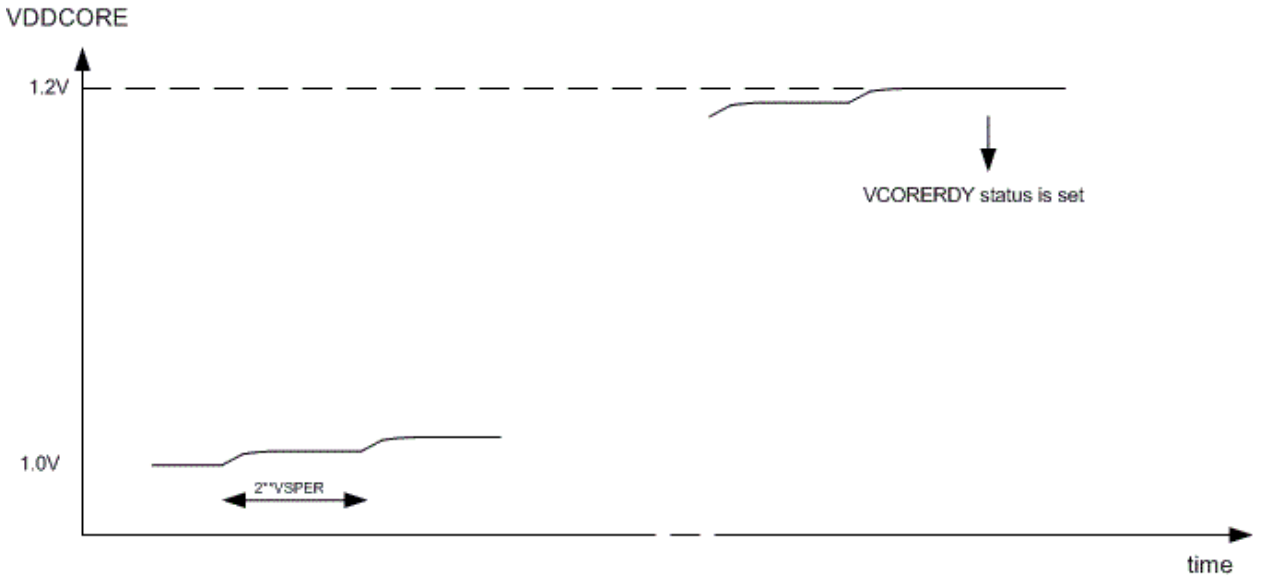
- When a Sleep mode (Standby, Hibernate, Backup) is entered or exited
- When a sleepwalking task is requested in Standby Sleep mode

To prevent high peak current on the main power supply and to have a smooth transition of VDDCORE, the Voltage Scaling Period field in VREG (VREG.VSPER) can be controlled: VDDCORE is changed by a typical 5 mV of the selected voltage scaling period ($2^{VSPER}$) * T until the target voltage is reached.

The smooth transition of VDDCORE is enabled/disabled by setting/clearing the Voltage Scaling Enable bit in VREG (VREG.VSEN).

The following waveform shows an example of exiting the Standby Sleep mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.




The STATUS.VCORERDY bit is set to '1' as soon as the VDDCORE voltage has reached the target voltage. During voltage transition, STATUS.VCORERDY will read '0'. The Voltage Ready interrupt (VCORERDY) can be used to detect a 0-to-1 transition of STATUS.VCORERDY, see also Interrupts.

When entering the Standby, Hibernate, or Backup Sleep mode, and when no sleepwalking task is requested, the VDDCORE Voltage scaling control is not used. This means that the voltage scale rise rate control (smooth transition) operates only when the device exits from sleep modes, and when the device is already in standby sleep mode and sleep walking is requested.

#### 19.6.1.5. Sleep Mode Operation

In Standby and Hibernate mode, the main voltage regulator (MAINVREG) operates in low power mode.

In backup mode, the low-power voltage regulator (LPVREG) is used to supply VDDCORE.

### 19.6.2. Voltage Reference System Operation

Different reference voltages are generated by the SUPC:

- A fixed-voltage source: BANDGAP=1.1V
- A variable voltage: VREF

#### 19.6.2.1. Initialization

The voltage reference output and the temperature sensor are disabled after any Reset.

#### 19.6.2.2. Enabling, Disabling, and Resetting

The voltage reference output is enabled/disabled by setting/clearing the Voltage Reference Output Enable bit in the Voltage Reference register (VREF.VREFOE).

The temperature sensor is enabled/disabled by setting/clearing the Temperature Sensor Enable bit in the Voltage Reference register (VREF.TSEN).

**Note:** When VREF.ONDEMAND=0, it is not recommended to enable both voltage reference output and temperature sensor at the same time - only the voltage reference output will be present at both ADC inputs.

#### 19.6.2.3. Selecting a Voltage Reference

The Voltage Reference Selection bit field in the VREF register (VREF.SEL) selects the voltage of VREF to be applied to analog modules, e.g. the ADC.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 19.6.2.4. Sleep Mode Operation

The Voltage Reference output and the Temperature Sensor output behavior during sleep mode can be configured using the Run in Standby bit and the On Demand bit in the Voltage Reference register (VREF.RUNSTDBY, VREF.ONDEMAND), refer to the table below:

**Table 19-1.** VREF Sleep Mode Operation

| VREF.ONDEMAND | VREF.RUNSTDBY | Voltage Reference Sleep behavior |
|---|---|---|
| 0 | 0 | Always run in all sleep modes *except* Standby Sleep mode |
| 0 | 1 | Always run in all sleep modes *including* Standby Sleep mode |
| 1 | 0 | Only run if requested by the ADC, in all sleep modes *except* Standby Sleep mode |
| 1 | 1 | Only run if requested by the ADC, in all sleep modes *including* Standby Sleep mode |

### 19.6.3. Battery Backup Power Switch

#### 19.6.3.1. Initialization

The Battery Backup Power Switch (BBPS) is disabled at power-up, and the backup domain is supplied by main power.

#### 19.6.3.2. Automatic Battery Backup Power Switch

The supply of the backup domain can be switched automatically to VBAT supply pin by the Battery Backup Power Switch when the BOD33 detects that the VDD supply is below the VDD threshold level (BOD33.LEVEL). It is switched back to VDD supply pin when the BOD33 detects that VDD is above the VDD threshold level (BOD33.LEVEL).

To enable this feature, the following configuration is required: BOD33.ACTION=BKUP.

#### 19.6.3.3. Sleep Mode Operation

The Battery Backup Power Switch is not stopped in any sleep mode.

##### 19.6.3.3.1. Entering Battery Backup Mode

Entering battery backup mode can be triggered by either:

- Wait-for-interrupt (WFI) instruction in case battery backup is forced in backup mode (BBPS.CONF=0x1)
- BOD33 detection: When the BOD33 detects loss of Main Power, the Backup Domain will be powered by battery and the device will enter the backup mode. For this trigger, the following register configuration is required: BOD33.ACTION=BKUP.

**Related Links**

PM – Power Manager

##### 19.6.3.3.2. Leaving Battery Backup Mode

Leaving battery backup mode is triggered by the RSTC when a Backup Mode Exit condition occurs. See RSTC module for details.

- BOD33 exit condition: When the BOD33 detects Main Power is restored and BOD33.ACTION=BKUP:
  - When BBPS.WAKEEN=1, the device will leave backup mode and wake up.
  - When BBPS.WAKEEN=0, the backup domain will be powered by Main Power, but the device will stay in backup mode.
- For other exit condition (RTC): The device is kept in battery-powered backup mode until Main Power is restored to supply the device. Then, the backup domain will be powered by Main Power.

### 19.6.4. Output Pins

The SUPC can drive two outputs. By writing a '1' to the corresponding Output Enable bit in the Backup Output Control register (BKOUT.EN), the OUTx pin is driven by the SUPC.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The OUT pin can be set by writing a '1' to the corresponding Set Output bit in the Backup Output Control register (BKOUT.SETx).

The OUT pin can be cleared by writing a '1' to the corresponding CLR bit (BKOUT.CLRx).

If a RTC Toggle Enable bit is written to '1' (BKOUT.RTCTGLx), the corresponding OUTx pin will toggle when an RTC event occurs.

### 19.6.5. Brown-Out Detectors

#### 19.6.5.1. Initialization

Before a Brown-Out Detector (BOD33) is enabled, it must be configured, as outlined by the following:

- Set the BOD threshold level (BOD33.LEVEL)
- Set the configuration in Active, Standby, Hibernate, and Backup modes (BOD33.ACTION, BOD33.STDBYCFG, BOD33.BKUP, BOD33.RUNHIB, and BOD33.RUNBKUP)
- Set the prescaling value if the BOD will run in sampling mode (BOD33.PSEL)
- Set the action and hysteresis (BOD33.ACTION and BOD33.HYST)

The BOD33 register is Enable-Protected, meaning that they can only be written when the BOD is disabled (BOD33.ENABLE=0 and STATUS.B33SRDY=0). As long as the Enable bit is '1', any writes to Enable-Protected registers will be discarded, and an APB error will be generated. The Enable bits are not Enable-Protected.

#### 19.6.5.2. Enabling, Disabling, and Resetting

After power on or user reset, the BOD33 register values are loaded from the NVM User Row.

The BOD33 is enabled by writing a '1' to the Enable bit in the BOD control register (BOD33.ENABLE). The BOD33 is disabled by writing a '0' to the BOD33.ENABLE.

**Related Links**

PM – Power Manager

NVM User Row Mapping (UROW)

#### 19.6.5.3. 3.3V Brown-Out Detector (BOD33)

The 3.3V Brown-Out Detector (BOD33) is able to monitor either the VDD or the VBAT supply and compares the voltage with the brown-out threshold levels.

In all mode except battery backup mode, the BOD33 compares the VDD voltage with the brown-out threshold level. This level is set in the BOD33 Level field in the BOD33 register (BOD33.LEVEL). When VDD crosses below the brown-out threshold level, the BOD33 can generate either an interrupt,or a Reset, or an Automatic Battery Backup Power Switch, depending on the BOD33 Action bit field (BOD33.ACTION).

In battery backup mode, the BOD33 monitors both the VBAT and VDD supplies alternatively. When VBAT crosses below the backup brown-out threshold level (BOD33.VBATLEVEL), the BOD33 generates a Power Supply Reset. When VDD crosses above the brown-out threshold level (BOD33.LEVEL), the device will leave battery backup mode and will wakeup from backup mode if the BBPS.WAKEEN bit is set.

The BOD33 detection status can be read from the BOD33 Detection bit in the Status register (STATUS.BOD33DET).

At start-up or at Power-On Reset (POR), the BOD33 register values are loaded from the NVM User Row.

**Related Links**

NVM User Row Mapping (UROW)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 19.6.5.3.1. BOD33 Sampling Mode

The Sampling Mode is a low-power mode where the BOD33 is being repeatedly enabled on a sampling clock's ticks. The BOD33 will monitor the supply voltage (VDD or VBAT) for a short period of time and then go to a low-power disabled state until the next sampling clock tick.

Sampling mode is enabled in Backup or Hibernate mode by writing to the BOD33 bits (BOD33.BKUPCFG = 1 or BOD33.HIBCFG = 1). The frequency of the clock ticks ($F_{clksampling}$) is controlled by the Prescaler Select bit groups in the BOD33 register (BOD33.PSEL).

$$F_{clksampling} = \frac{F_{clkprescaler}}{2^{(\mathrm{PSEL}+1)}}$$

The prescaler signal ($F_{clkprescaler}$) is a 32.768 kHz clock, output by the 32 kHz Ultra Low-Power Oscillator OSCULP32K.

**Note:** If (BOD33.PSEL) is 0, sampling mode is disabled.

As the sampling clock is different from the APB clock domain, synchronization among the clocks is necessary. See Synchronization for additional information.

**Related Links**

NVM User Row Mapping (UROW)

#### 19.6.5.3.2. BOD33 Low Power Mode

BOD33 Low Power mode is automatically enabled in Backup or Hibernate sleep mode.

BOD33 Low Power mode can be enabled in Standby sleep mode by writting to '1' the BOD33.STDBYCFG bit.

**Related Links**

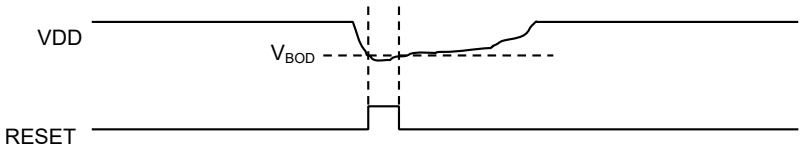
NVM User Row Mapping (UROW)

#### 19.6.5.3.3. BOD33 Hysteresis

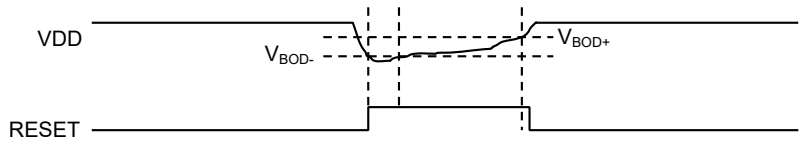
A hysteresis on the trigger threshold of a BOD will reduce the sensitivity to ripples on the monitored voltage: instead of switching $\overline{\text{RESET}}$ at each crossing of $V_{BOD}$, the thresholds for switching $\overline{\text{RESET}}$ on and off are separated ($V_{BOD-}$ and $V_{BOD+}$, respectively).

**Figure 19-2.** BOD Hysteresis Principle

Hysteresis OFF:




Hysteresis ON:




Enabling the BOD33 hysteresis by writing the Hysteresis bit field in the BOD33 register (BOD33.HYST) to a non-null value will add hysteresis to the BOD33 threshold level.

The hysteresis functionality can be used in Sampling Mode.

**Related Links**

NVM User Row Mapping (UROW)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 19.6.5.3.4. Standby Sleep Mode

The BOD33 can be used in standby mode if the BOD is enabled and the Run in Standby bit is written to '1' (BOD33.RUNSTDBY).

It is set in Low Power mode if the BOD33.STDBYCFG bit is written to '1'.

**Related Links**

NVM User Row Mapping (UROW)

#### 19.6.5.3.5. Backup and Hibernate sleep Modes

To enable the BOD33 in Backup or Hibernate sleep mode, the Run in Backup or Hibernate sleep mode bits in the BOD33 register (BOD33.RUNBKUP, BOD33.RUNHIB) must be written to '1'. The BOD33 is automatically set in BOD33 Ultra Low-Power mode. Additionnaly, the BOD33 will operate in Sampling mode if the BOD33.PSEL bit is non-null. In this state, the voltage monitored by BOD33 is always the supply of the backup domain, i.e. VDD or VBAT.

**Related Links**

NVM User Row Mapping (UROW)

### 19.6.5.4. 1.2V Brown-Out Detector (BOD12)

The BOD12 is calibrated in production and its calibration configuration is stored in the NVM User Row. This configuration must not be changed to assure the correct behavior of the BOD12. The BOD12 generates a reset when 1.2V crosses below the preset brown-out level. The BOD12 is always disabled in Standby, Hibernate, and Backup Sleep modes.

**Related Links**

NVM User Row Mapping (UROW)

## 19.6.6. Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- VDDCORE Voltage Ready (VCORERDY), asynchronous
- Voltage Regulator Ready (VREGRDY) asynchronous
- BOD33 Ready (BOD33RDY), synchronous
- BOD33 Detection (BOD33DET), asynchronous
- BOD33 Synchronization Ready (B33SRDY), synchronous

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the SUPC is reset. See the INTFLAG register for details on how to clear interrupt flags. The user must read the INTFLAG register to determine which interrupt condition is present.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

## 19.6.7. Synchronization

The prescaler counters that are used to trigger brown-out detections operate asynchronously from the peripheral bus. As a consequence, the BOD33 Enable bit (BOD33.ENABLE) need synchronization when written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The Write-Synchronization of the Enable bit is triggered by writing a '1' to the Enable bit of the BOD33 Control register. The Synchronization Ready bit (STATUS.B33SRDY) in the STATUS register will be cleared when the Write-Synchronization starts, and set again when the Write-Synchronization is complete. Writing to the same register while the Write-Synchronization is ongoing (STATUS.B33SRDY is '0') will generate a PAC error without stalling the APB bus.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 19.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | INTENCLR | 7:0 | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| | | 15:8 | | | | | | VCORERDY | | VREGRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | INTENSET | 7:0 | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| | | 15:8 | | | | | | VCORERDY | | VREGRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | INTFLAG | 7:0 | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| | | 15:8 | | | | | | VCORERDY | | VREGRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | STATUS | 7:0 | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| | | 15:8 | | | | | | VCORERDY | | VREGRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | BOD33 | 7:0 | RUNBKUP | RUNHIB | RUNSTDBY | STDBYCFG | ACTION[1:0] | | ENABLE | |
| | | 15:8 | | PSEL[2:0] | | | HYST[3:0] | | | |
| | | 23:16 | LEVEL[7:0] | | | | | | | |
| | | 31:24 | VBATLEVEL[7:0] | | | | | | | |
| 0x14 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | VREG | 7:0 | RUNBKUP | | | | | SEL | ENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | VSEN |
| | | 31:24 | | | | | | VSPER[2:0] | | |
| 0x1C | VREF | 7:0 | ONDEMAND | RUNSTDBY | | | TSSEL | VREFOE | TSEN | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | SEL[3:0] | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | BBPS | 7:0 | | | | | | WAKEEN | | CONF |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | BKOUT | 7:0 | | | | | | | EN1 | EN0 |
| | | 15:8 | | | | | | | CLR1 | CLR0 |
| | | 23:16 | | | | | | | SET1 | SET0 |
| | | 31:24 | | | | | | | RTCTGL1 | RTCTGL0 |
| 0x28 | BKIN | 7:0 | | | | | | | BKIN1 | BKIN0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 19.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). PAC Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to Register Access Protection for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. Refer to Synchronization for details.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 19.8.1. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | VCORERDY | | VREGRDY |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 10 – VCORERDY** VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the VDDCORE Ready Interrupt Enable bit, which disables the VDDCORE Ready interrupt.

| Value | Description |
|---|---|
| 0 | The VDDCORE Ready interrupt is disabled. |
| 1 | The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set. |

**Bit 8 – VREGRDY** Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Voltage Regulator Ready Interrupt Enable bit, which disables the Voltage Regulator Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Voltage Regulator Ready interrupt is disabled. |
| 1 | The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set. |

**Bit 2 – B33SRDY** BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Synchronization Ready Interrupt Enable bit, which disables the BOD33 Synchronization Ready interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The BOD33 Synchronization Ready interrupt is disabled. |
| 1 | The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set. |

**Bit 1 – BOD33DET** BOD33 Detection Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Detection Interrupt Enable bit, which disables the BOD33 Detection interrupt.

| Value | Description |
|---|---|
| 0 | The BOD33 Detection interrupt is disabled. |
| 1 | The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set. |

**Bit 0 – BOD33RDY** BOD33 Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the BOD33 Ready Interrupt Enable bit, which disables the BOD33 Ready interrupt.

| Value | Description |
|---|---|
| 0 | The BOD33 Ready interrupt is disabled. |
| 1 | The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 19.8.2. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | VCORERDY | | VREGRDY |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 10 – VCORERDY** VDDCORE Voltage Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the VDDCORE Ready Interrupt Enable bit, which enables the VDDCORE Ready interrupt.

| Value | Description |
|---|---|
| 0 | The VDDCORE Ready interrupt is disabled. |
| 1 | The VDDCORE Ready interrupt is enabled and an interrupt request will be generated when the VCORERDY Interrupt Flag is set. |

**Bit 8 – VREGRDY** Voltage Regulator Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Voltage Regulator Ready Interrupt Enable bit, which enables the Voltage Regulator Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Voltage Regulator Ready interrupt is disabled. |
| 1 | The Voltage Regulator Ready interrupt is enabled and an interrupt request will be generated when the Voltage Regulator Ready Interrupt Flag is set. |

**Bit 2 – B33SRDY** BOD33 Synchronization Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Synchronization Ready Interrupt Enable bit, which enables the BOD33 Synchronization Ready interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The BOD33 Synchronization Ready interrupt is disabled. |
| 1 | The BOD33 Synchronization Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Synchronization Ready Interrupt flag is set. |

**Bit 1 – BOD33DET** BOD33 Detection Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Detection Interrupt Enable bit, which enables the BOD33 Detection interrupt.

| Value | Description |
|---|---|
| 0 | The BOD33 Detection interrupt is disabled. |
| 1 | The BOD33 Detection interrupt is enabled, and an interrupt request will be generated when the BOD33 Detection Interrupt flag is set. |

**Bit 0 – BOD33RDY** BOD33 Ready Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the BOD33 Ready Interrupt Enable bit, which enables the BOD33 Ready interrupt.

| Value | Description |
|---|---|
| 0 | The BOD33 Ready interrupt is disabled. |
| 1 | The BOD33 Ready interrupt is enabled, and an interrupt request will be generated when the BOD33 Ready Interrupt flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 19.8.3. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x08
**Reset:** 0x0000010X
**Property:** -

In the reset value: X= determined from NVM User Row (0xX=0bx00y)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | VCORERDY | | VREGRDY |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | y |

**Bit 10 – VCORERDY** VDDCORE Voltage Ready
This flag is cleared by writing a '1 to it.
This flag is set on a zero-to-one transition of the VDDCORE Ready bit in the Status register (STATUS.VCORERDY) and will generate an interrupt request if INTENSET.VCORERDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the VCORERDY interrupt flag.

**Bit 8 – VREGRDY** Voltage Regulator Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the Voltage Regulator Ready bit in the Status register (STATUS.VREGRDY) and will generate an interrupt request if INTENSET.VREGRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the VREGRDY interrupt flag.

**Bit 2 – B33SRDY** BOD33 Synchronization Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Synchronization Ready bit in the Status register (STATUS.B33SRDY) and will generate an interrupt request if INTENSET.B33SRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Synchronization Ready interrupt flag.

**Bit 1 – BOD33DET** BOD33 Detection
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Detection bit in the Status register (STATUS.BOD33DET) and will generate an interrupt request if INTENSET.BOD33DET=1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Detection interrupt flag.

**Bit 0 – BOD33RDY** BOD33 Ready

This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the BOD33 Ready bit in the Status register (STATUS.BOD33RDY) and will generate an interrupt request if INTENSET.BOD33RDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the BOD33 Ready interrupt flag.
The BOD33 can be enabled.

**Related Links**

NVM User Row Mapping (UROW)


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 19.8.4. Status

**Name:** STATUS
**Offset:** 0x0C
**Reset:** Determined from NVM User Row
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | VCORERDY | | VREGRDY |
| Access | | | | | | R | | R |
| Reset | | | | | | 1 | | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | B33SRDY | BOD33DET | BOD33RDY |
| Access | | | | | | R | R | R |
| Reset | | | | | | 1 | 0 | y |

**Bit 10 – VCORERDY** VDDCORE Voltage Ready

| Value | Description |
|---|---|
| 0 | the VDDCORE voltage is not as expected. |
| 1 | the VDDCORE voltage is the target voltage. |

**Bit 8 – VREGRDY** Voltage Regulator Ready

| Value | Description |
|---|---|
| 0 | The selected voltage regulator in VREG.SEL is not ready. |
| 1 | The voltage regulator selected in VREG.SEL is ready and the core domain is supplied by this voltage regulator. |

**Bit 2 – B33SRDY** BOD33 Synchronization Ready

| Value | Description |
|---|---|
| 0 | BOD33 synchronization is ongoing. |
| 1 | BOD33 synchronization is complete. |

**Bit 1 – BOD33DET** BOD33 Detection

| Value | Description |
|---|---|
| 0 | No BOD33 detection. |
| 1 | BOD33 has detected that the I/O power supply is going below the BOD33 reference value. |

**Bit 0 – BOD33RDY** BOD33 Ready
The BOD33 can be enabled at start-up from NVM User Row.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | BOD33 is not ready. |
| 1 | BOD33 is ready. |

**Related Links**

NVM User Row Mapping (UROW)


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 19.8.5. 3.3V Brown-Out Detector (BOD33) Control

**Name:** BOD33
**Offset:** 0x10
**Reset:** Determined from NVM User Row
**Property:** Write-Synchronized, PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | VBATLEVEL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | LEVEL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | PSEL[2:0] | | | HYST[3:0] | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNBKUP | RUNHIB | RUNSTDBY | STDBYCFG | ACTION[1:0] | | ENABLE | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | y | y | z | |

**Bits 31:24 – VBATLEVEL[7:0]** BOD33 Threshold Level on VBAT
This field sets the triggering voltage threshold for the BOD33 when the BOD33 monitors VBAT in battery backup sleep mode.
This field is not synchronized.

**Bits 23:16 – LEVEL[7:0]** BOD33 Threshold Level on VDD
This field sets the triggering voltage threshold for the BOD33 when the BOD33 monitors VDD. If an hysteresis value is programmed (BOD33.HYST), this field corresponds to the lower threshold ($V_{BOD-}$).
These bits are loaded from NVM User Row at start-up.
This field is not synchronized.
The VBOD- input voltage can be calculated as follows: VBOD- = 1.5 + LEVEL[7:0) x $VBOD33_{LEVEL_STEP}$
And the upper threshold (VBOD+) is then: VBOD+ = VBOD- + N x $VBOD33_{HYST_STEP}$, With N=0 to 15 according to HYST[3:0] value and $VBOD33_{HYST_STEP}$ = $VBOD33_{LEVEL_STEP}$, (refer to Bits 11:8 – HYST[3:0]: BOD33 Hysteresis voltage value on VDD).
If a hysteresis value is programmed, the following equation must be respected to avoid any overflow on VBOD33(max):
LEVEL[7:0] value < 255 - HYST[3:0] value.

**Bits 14:12 – PSEL[2:0]** Prescaler Select
Selects the prescaler divide-by output for the BOD33 sampling mode available in hibernate, backup or battery backup mode. The input clock comes from the OSCULP32K 32.768 kHz output.

| Value | Name | Description |
|---|---|---|
| 0x0 | NODIV | Not divided: Sampling mode is OFF. |
| 0x1 | DIV4 | Divide clock by 4 |
| 0x2 | DIV8 | Divide clock by 8 |
| 0x3 | DIV16 | Divide clock by 16 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x4 | DIV32 | Divide clock by 32 |
| 0x5 | DIV64 | Divide clock by 64 |
| 0x6 | DIV128 | Divide clock by 128 |
| 0x7 | DIV256 | Divide clock by 256 |

**Bits 11:8 – HYST[3:0]** BOD33 Hysteresis Voltage Value on $V_{DD}$
This field sets the hysteresis voltage value related to "BOD33 Threshold Level on VDD" field when the BOD33 monitors VDD.
These bits are loaded from NVM User Row at start-up.
This field is not synchronized.

| Value | Description |
|---|---|
| 0 | No hysteresis. |
| N | Hysteresis value is set to N*HYST_STEP.<br>See the Electrical Characteristics section for the HYST_STEP voltage level. |

**Bit 7 – RUNBKUP** BOD33 Configuration in Backup Sleep Mode
This field is not synchronized.

| Value | Description |
|---|---|
| 0 | In backup sleep mode, the BOD33 is disabled. |
| 1 | In backup sleep mode, the BOD33 is enabled and configured in sampling mode. |

**Bit 6 – RUNHIB** BOD33 Configuration in Hibernate Sleep Mode
This field is not synchronized.

| Value | Description |
|---|---|
| 0 | In hibernate sleep mode, the BOD33 is disabled. |
| 1 | In hibernate sleep mode, the BOD33 is enabled and configured in sampling mode. |

**Bit 5 – RUNSTDBY** Run in Standby
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | In standby sleep mode, the BOD33 is disabled. |
| 1 | In standby sleep mode, the BOD33 is enabled. |

**Bit 4 – STDBYCFG** BOD33 Configuration in Standby Sleep Mode
If the RUNSTDBY bit is set to '1', the STDBYCFG bit sets the BOD33 configuration in standby sleep mode.
This field is not synchronized.

| Value | Description |
|---|---|
| 0 | In standby sleep mode, the BOD33 is enabled and configured in normal mode. |
| 1 | In standby sleep mode, the BOD33 is enabled and configured in low power mode. |

**Bits 3:2 – ACTION[1:0]** BOD33 Action
These bits are used to select the BOD33 action when the supply voltage crosses below the BOD33 threshold.
These bits are loaded from NVM User Row at start-up.
This field is not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | RESET | The BOD33 generates a reset |
| 0x2 | INT | The BOD33 generates an interrupt |
| 0x3 | BKUP | The BOD33 puts the device in battery backup sleep mode. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 1 – ENABLE** Enable

This bit is loaded from NVM User Row at start-up.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | BOD33 is disabled. |
| 1 | BOD33 is enabled. |

**Related Links**

NVM User Row Mapping (UROW)


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 19.8.6. Voltage Regulator System (VREG) Control

**Name:** VREG
**Offset:** 0x18
**Reset:** 0x00000002
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | VSPER[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | VSEN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNBKUP | | | | | SEL | ENABLE | |
| Access | R/W | | | | | R/W | R/W | |
| Reset | 0 | | | | | 0 | 1 | |

**Bits 26:24 – VSPER[2:0]** Voltage Scaling Period
This bitfield defines the time between the voltage steps when the VDDCORE voltage scaling is enabled.
The time is $(2^{VSPER})$ * T, where T is an internal period (typ 250 ns).

**Bit 16 – VSEN** Voltage Scaling Enable

| Value | Description |
|---|---|
| 0 | The voltage scaling is disabled. |
| 1 | The voltage scaling is enabled. |

**Bit 7 – RUNBKUP** Run in Backup
This bit controls how the main voltage regulator behaves in backup sleep mode.

| Value | Description |
|---|---|
| 0 | The main voltage regulator is halted during backup sleep mode. |
| 1 | The main voltage regulator is not stopped during backup sleep mode. |

**Bit 2 – SEL** Voltage Regulator Selection
This bit is loaded from NVM User Row at start-up. Refer to *NVM User Row Mapping* section for more details.

| Value | Description |
|---|---|
| 0 | The main voltage regulator is a LDO voltage regulator. |
| 1 | The main voltage regulator is a buck converter. |

**Bit 1 – ENABLE** Must be set to 1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
NVM User Row Mapping (UROW)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 19.8.7. Voltage References System (VREF) Control

**Name:** VREF
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SEL[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | | TSSEL | VREFOE | TSEN | |
| Access | R/W | R/W | | | R/W | R/W | R/W | |
| Reset | 0 | 0 | | | 0 | 0 | 0 | |

**Bits 19:16 – SEL[3:0]** Voltage Reference Selection
These bits select the Voltage Reference for the ADC/DAC.

| Value | Name | Description |
|---|---|---|
| 0x0 | 1V0 | 1.0V voltage reference typical value |
| 0x1 | 1V1 | 1.1V voltage reference typical value |
| 0x2 | 1V2 | 1.2V voltage reference typical value |
| 0x3 | 1V25 | 1.25V voltage reference typical value |
| 0x4 | 2V0 | 2.0V voltage reference typical value |
| 0x5 | 2V2 | 2.2V voltage reference typical value |
| 0x6 | 2V4 | 2.4V voltage reference typical value |
| 0x7 | 2V5 | 2.5V voltage reference typical value |
| Others | | Reserved |

**Bit 7 – ONDEMAND** On Demand Control
The On Demand operation mode allows to enable or disable the voltage reference depending on peripheral requests.

| Value | Description |
|---|---|
| 0 | The voltage reference is always on, if enabled. |
| 1 | The voltage reference is enabled when a peripheral is requesting it. The voltage reference is disabled if no peripheral is requesting it. |

**Bit 6 – RUNSTDBY** Run In Standby
The bit controls how the voltage reference behaves during standby sleep mode.

| Value | Description |
|---|---|
| 0 | The voltage reference is halted during standby sleep mode. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The voltage reference is not stopped in standby sleep mode. If VREF.ONDEMAND=1, the voltage reference will be running when a peripheral is requesting it. If VREF.ONDEMAND=0, the voltage reference will always be running in standby sleep mode. |

**Bit 3 – TSSEL** Temperature Sensor Channel Selection

| Value | Description |
|---|---|
| 0 | The Temperature Sensor PTAT channel is selected. |
| 1 | The Temperature Sensor CTAT channel is selected. |

**Bit 2 – VREFOE** Voltage Reference Output Enable

| Value | Description |
|---|---|
| 0 | The Voltage Reference output (INTREF) is not available as an ADC input channel. |
| 1 | The Voltage Reference output (INTREF) is routed to an ADC input channel. |

**Bit 1 – TSEN** Temperature Sensor Enable

| Value | Description |
|---|---|
| 0 | Temperature Sensor is disabled. |
| 1 | Temperature Sensor is enabled and routed to an ADC input channel. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 19.8.8. Battery Backup Power Switch (BBPS) Control

**Name:** BBPS
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WAKEEN | | CONF |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – WAKEEN** Wake Enable

| Value | Description |
|---|---|
| 0 | The device is not woken up when switched from battery backup power to Main Power. |
| 1 | The device is woken up when switched from battery backup power to Main Power. |

**Bit 0 – CONF** Battery Backup Power Switch Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | BOD33 | The power switch is handled by the BOD33 according to the BOD33.ACTION bit field. |
| 0x1 | FORCED | In backup sleep mode, the backup domain is always supplied by Battery Backup Power. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 19.8.9. Backup Output (BKOUT) Control

**Name:** BKOUT
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RTCTGL1 | RTCTGL0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | SET1 | SET0 |
| Access | | | | | | | W | W |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CLR1 | CLR0 |
| Access | | | | | | | W | W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EN1 | EN0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 24, 25 – RTCTGL** RTC Toggle Output

| Value | Description |
|---|---|
| 0 | The output will not toggle on RTC event. |
| 1 | The output will toggle on RTC event. |

**Bits 16, 17 – SET** Set Output
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will set the corresponding output.
Reading this bit returns '0'.

**Bits 8, 9 – CLR** Clear Output
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will clear the corresponding output.
Reading this bit returns '0'.

**Bits 0, 1 – EN** Enable Output

| Value | Description |
|---|---|
| 0 | The output is not enabled. |
| 1 | The output is enabled and driven by the SUPC. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 19.8.10. Backup Input (BKIN) Value

**Name:** BKIN
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BKIN1 | BKIN0 |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bits 0, 1 – BKIN** Backup Input Value
These bits are cleared when the corresponding backup I/O pin detects a logical low level on the input pin or when the backup I/O is not enabled.
These bits are set when the corresponding backup I/O pin detects a logical high level on the input pin when the backup I/O is enabled.

| Value | Name | Description |
|---|---|---|
| BKIN[0] | OUT[0] | If BKOUT.EN[0]=1, BKIN[0] will give the input value of the OUT[0] pin |
| BKIN[1] | OUT[1] | If BKOUT.EN[1]=1, BKIN[1] will give the input value of the OUT[1] pin |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 20. WDT – Watchdog Timer

## 20.1. Overview

The Watchdog Timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is configured to a predefined time-out period, and is constantly running when enabled. If the WDT is not cleared within the time-out period, it will issue a system reset. An early-warning interrupt is available to indicate an upcoming watchdog time-out condition.

The window mode makes it possible to define a time slot (or window) inside the total time-out period during which the WDT must be cleared. If the WDT is cleared outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared too frequently.

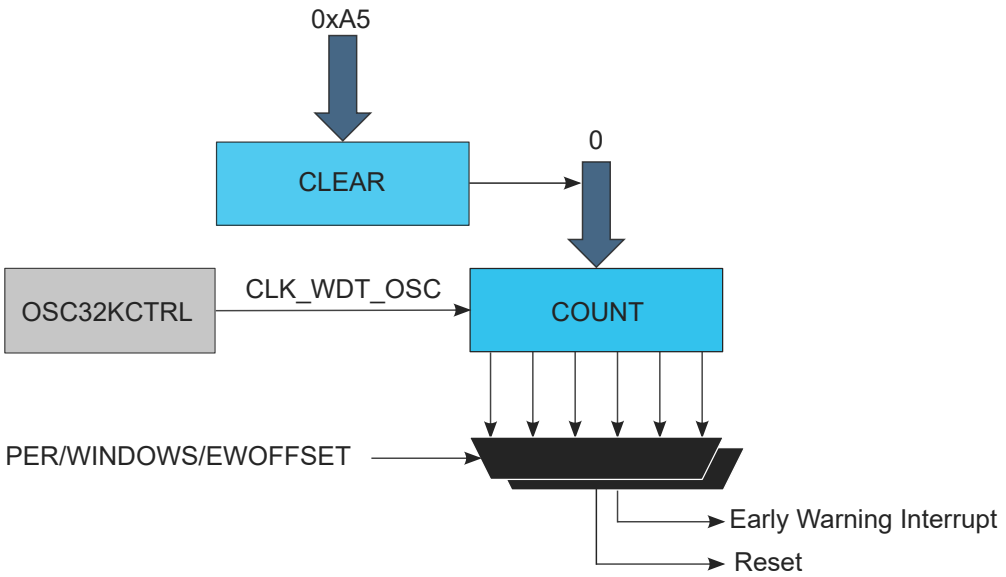
When enabled, the WDT will run in active mode and any sleep modes, except Hibernate, Backup and OFF sleep mode. It is asynchronous and runs from a CPU-independent clock source (OSCULP32K). The WDT will continue operation and issue a system reset or interrupt even if the main clocks fail.

## 20.2. Features

- Issues a system reset if the Watchdog Timer is not cleared before its time-out period
- Early Warning interrupt generation
- Asynchronous operation from dedicated oscillator
- Two types of operation
  - Normal
  - Window mode
- Selectable time-out periods
  - From 8 cycles to 16,384 cycles in Normal mode
  - From 16 cycles to 32,768 cycles in Window mode
- Always-On capability

## 20.3. Block Diagram

**Figure 20-1.** WDT Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 20.4. Signal Description

Not applicable.

## 20.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 20.5.1. I/O Lines

Not applicable.

### 20.5.2. Power Management

The WDT can continue to operate in any sleep modes where the selected source clock is running. The WDT interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

**Related Links**

PM – Power Manager

### 20.5.3. Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the Main Clock module (MCLK).

A 1.024 kHz clock (CLK_WDT_OSC) is used to clock the WDT internal counter from the internal Ultra Low-Power Oscillator (OSCULP32K).

Due to ultra low-power design, the oscillator is not very accurate, therefore the exact time-out period may vary from device-to-device. This variation must be considered when designing software that uses the WDT to ensure that the time-out periods used are valid for all devices.

The counter clock CLK_WDT_OSC is asynchronous to the bus clock (CLK_WDT_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**

Peripheral Clock Masking
OSC32KCTRL – 32KHz Oscillators Controller

### 20.5.4. DMA

Not applicable.

### 20.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the WDT interrupt(s) requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 20.5.6. Events

Not applicable.

### 20.5.7. Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

### 20.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Interrupt Flag Status and Clear (INTFLAG) register

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

### 20.5.9. Analog Connections

Not applicable.

## 20.6. Functional Description

### 20.6.1. Principle of Operation

The Watchdog Timer (WDT) is a system for monitoring correct program operation, making it possible to recover from error situations such as runaway code, by issuing a Reset. When enabled, the WDT is a constantly running timer that is configured to a predefined time-out period. Before the end of the time-out period, the WDT should be set back, or else, a system Reset is issued.

The WDT has two modes of operation, Normal mode and Window mode. Both modes offer the option of Early Warning interrupt generation. The description for each of the basic modes is given below. The settings in the Control A register (CTRLA) and the Interrupt Enable register (handled by INTENCLR/INTENSET) determine the mode of operation:

**Table 20-1.** WDT Operating Modes

| CTRLA.ENABLE | CTRLA.WEN | Interrupt Enable | Mode |
|---|---|---|---|
| 0 | x | x | Stopped |
| 1 | 0 | 0 | Normal mode |
| 1 | 0 | 1 | Normal mode with Early Warning interrupt |
| 1 | 1 | 0 | Window mode |
| 1 | 1 | 1 | Window mode with Early Warning interrupt |

### 20.6.2. Basic Operation

#### 20.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the WDT is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except the Enable bit (CTRLA.ENABLE)
- Configuration register (CONFIG)
- Early Warning Interrupt Control register (EWCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

The WDT can be configured only while the WDT is disabled. The WDT is configured by defining the required Time-Out Period bits in the Configuration register (CONFIG.PER). If Window mode operation is desired, the Window Enable bit in the Control A register must be set (CTRLA.WEN=1) and the Window Period bits in the Configuration register (CONFIG.WINDOW) must be defined.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

#### 20.6.2.2. Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the NVM User Row.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Watchdog Timer Windows Mode Enable bit in the Control A register, CTRLA.WEN
- Watchdog Timer Windows Mode Time-Out Period bits in the Configuration register, CONFIG.WINDOW
- Time-Out Period bits in the Configuration register, CONFIG.PER
- Early Warning Interrupt Time Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET

### 20.6.2.3. Enabling, Disabling, and Resetting

The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The WDT is disabled by writing a '0' to CTRLA.ENABLE.

The WDT can be disabled only if the Always-On bit in the Control A register (CTRLA.ALWAYSON) is '0'.

### 20.6.2.4. Normal Mode

In Normal mode operation, the length of a time-out period is configured in CONFIG.PER. The WDT is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). Once enabled, the WDT will issue a system reset if a time-out occurs. This can be prevented by clearing the WDT at any time during the time-out period.

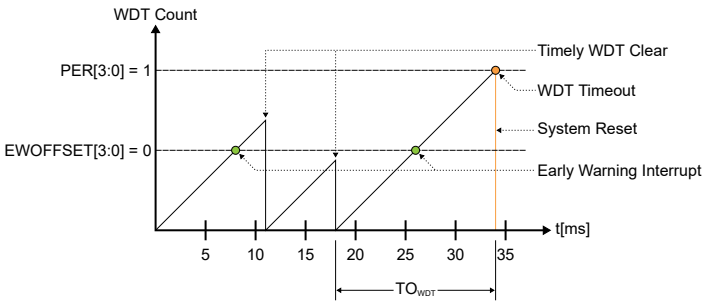
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

There are 12 possible WDT time-out ($TO_{WDT}$) periods, selectable from 8ms to 16s.

By default, the early warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear register (INTENCLR.EW).

If the Early Warning Interrupt is enabled, an interrupt is generated prior to a WDT time-out condition. In Normal mode, the Early Warning Offset bits in the Early Warning Interrupt Control register, EWCTRL.EWOFFSET, define the time when the early warning interrupt occurs. The Normal mode operation is illustrated in the figure Normal-Mode Operation.

**Figure 20-2.** Normal-Mode Operation




### 20.6.2.5. Window Mode

In Window mode operation, the WDT uses two different time specifications: the WDT can only be cleared by writing 0xA5 to the CLEAR register *after* the closed window time-out period ($TO_{WDTW}$), during the subsequent Normal time-out period ($TO_{WDT}$). If the WDT is cleared before the time window opens (before $TO_{WDTW}$ is over), the WDT will issue a system reset.

Both parameters $TO_{WDTW}$ and $TO_{WDT}$ are periods in a range from 8ms to 16s, so the total duration of the WDT time-out period is the sum of the two parameters.

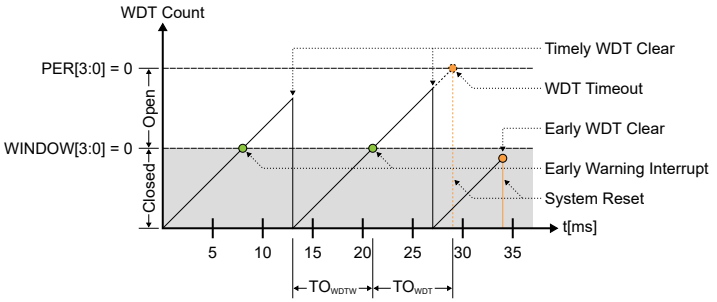

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable register (INTENSET.EW) must be written to '1'. The Early Warning Interrupt is disabled again by writing a '1' to the Early Warning Interrupt bit in the Interrupt Enable Clear (INTENCLR.EW) register.

If the Early Warning interrupt is enabled in Window mode, the interrupt is generated at the start of the open window period, i.e. after $TO_{WDTW}$. The Window mode operation is illustrated in figure Window-Mode Operation.

**Figure 20-3.** Window-Mode Operation




### 20.6.3. DMA Operation

Not applicable.

### 20.6.4. Interrupts

The WDT has the following interrupt source:

- Early Warning (EW): Indicates that the counter is approaching the time-out condition.
  - This interrupt is an asynchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the WDT is reset. See the INTFLAG register description for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

**Related Links**

Nested Vector Interrupt Controller
PM – Power Manager

### 20.6.5. Events

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.6.6. Sleep Mode Operation

The WDT will continue to operate in any sleep mode where the source clock is active except backup mode. The WDT interrupts can be used to wake up the device from a sleep mode. An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering an interrupt. In this case, the CPU will continue executing from the instruction following the entry into sleep.

**Related Links**

CTRLA

Control A

### 20.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following registers are synchronized when written:

- Enable bit in Control A register (CTRLA.ENABLE)
- Window Enable bit in Control A register (CTRLA.WEN)
- Always-On bit in control Control A (CTRLA.ALWAYSON)
- Watchdog Clear register (CLEAR)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

### 20.6.8. Additional Features

#### 20.6.8.1. Always-On Mode

The Always-On mode is enabled by setting the Always-On bit in the Control A register (CTRLA.ALWAYSON=1). When the Always-On mode is enabled, the WDT runs continuously, regardless of the state of CTRLA.ENABLE. Once written, the Always-On bit can only be cleared by a power-on reset. The Configuration (CONFIG) and Early Warning Control (EWCTRL) registers are read-only registers while the CTRLA.ALWAYSON bit is set. Thus, the time period configuration bits (CONFIG.PER, CONFIG.WINDOW, EWCTRL.EWOFFSET) of the WDT cannot be changed.

Enabling or disabling Window mode operation by writing the Window Enable bit (CTRLA.WEN) is allowed while in Always-On mode, but note that CONFIG.PER cannot be changed.

The Interrupt Clear and Interrupt Set registers are accessible in the Always-On mode. The Early Warning interrupt can still be enabled or disabled while in the Always-On mode, but note that EWCTRL.EWOFFSET cannot be changed.

Table WDT Operating Modes With Always-On shows the operation of the WDT for CTRLA.ALWAYSON=1.

**Table 20-2.** WDT Operating Modes With Always-On

| WEN | Interrupt Enable | Mode |
|---|---|---|
| 0 | 0 | Always-on and normal mode |
| 0 | 1 | Always-on and normal mode with Early Warning interrupt |
| 1 | 0 | Always-on and window mode |
| 1 | 1 | Always-on and window mode with Early Warning interrupt |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 20.6.8.2. Early Warning

The Early Warning interrupt notifies that the WDT is approaching its time-out condition. The Early Warning interrupt behaves differently in Normal mode and in Window mode.

*In Normal mode*, the Early Warning interrupt generation is defined by the Early Warning Offset in the Early Warning Control register (EWCTRL.EWOFFSET). The Early Warning Offset bits define the number of CLK_WDT_OSC clocks before the interrupt is generated, relative to the start of the watchdog time-out period.

The user must take caution when programming the Early Warning Offset bits. If these bits define an Early Warning interrupt generation time greater than the watchdog time-out period, the watchdog time-out system reset is generated prior to the Early Warning interrupt. Consequently, the Early Warning interrupt will never be generated.

*In window mode*, the Early Warning interrupt is generated at the start of the open window period. In a typical application where the system is in sleep mode, the Early Warning interrupt can be used to wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

**Example:** If the WDT is operating in Normal mode with CONFIG.PER = 0x2 and EWCTRL.EWOFFSET = 0x1, the Early Warning interrupt is generated 16 CLK_WDT_OSC clock cycles after the start of the time-out period. The time-out system reset is generated 32 CLK_WDT_OSC clock cycles after the start of the watchdog time-out period.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 20.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ALWAYSON | | | | | WEN | ENABLE | |
| 0x01 | CONFIG | 7:0 | WINDOW[3:0] | | | | PER[3:0] | | | |
| 0x02 | EWCTRL | 7:0 | | | | | EWOFFSET[3:0] | | | |
| 0x03 | Reserved | | | | | | | | | |
| 0x04 | INTENCLR | 7:0 | | | | | | | | EW |
| 0x05 | INTENSET | 7:0 | | | | | | | | EW |
| 0x06 | INTFLAG | 7:0 | | | | | | | | EW |
| 0x07 | Reserved | | | | | | | | | |
| 0x08 | SYNCBUSY | 7:0 | | | | CLEAR | ALWAYSON | WEN | ENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | CLEAR | 7:0 | CLEAR[7:0] | | | | | | | |

## 20.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** x initially determined from NVM User Row after reset
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ALWAYSON | | | | | WEN | ENABLE | |
| Access | R/W | | | | | R/W | R/W | |
| Reset | x | | | | | x | x | |

**Bit 7 – ALWAYSON** Always-On
This bit allows the WDT to run continuously. After being set, this bit cannot be written to '0', and the WDT will remain enabled until a power-on Reset is received. When this bit is '1', the Control A register (CTRLA), the Configuration register (CONFIG) and the Early Warning Control register (EWCTRL) will be read-only, and any writes to these registers are not allowed.
Writing a '0' to this bit has no effect.
This bit is not Enable-Protected.
This bit is loaded from NVM User Row at start-up.

| Value | Description |
|---|---|
| 0 | The WDT is enabled and disabled through the ENABLE bit. |
| 1 | The WDT is enabled and can only be disabled by a power-on reset (POR). |

**Bit 2 – WEN** Watchdog Timer Window Mode Enable
This bit enables Window mode. It can only be written if the peripheral is disabled unless CTRLA.ALWAYSON=1.
This bit is loaded from NVM User Row at startup.

| Value | Description |
|---|---|
| 0 | Window mode is disabled (normal operation). |
| 1 | Window mode is enabled. |

**Bit 1 – ENABLE** Enable
This bit enables or disables the WDT. It can only be written if CTRLA.ALWAYSON=0.
Due to synchronization, there is delay between writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not Enable-Protected.
This bit is loaded from NVM User Row at startup.

| Value | Description |
|---|---|
| 0 | The WDT is disabled. |
| 1 | The WDT is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.8.2. Configuration

**Name:** CONFIG
**Offset:** 0x01
**Reset:** x initially determined from NVM User Row after reset
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WINDOW[3:0] | | | | PER[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

**Bits 7:4 – WINDOW[3:0]** Window Mode Time-Out Period
In Window mode, these bits determine the watchdog closed window period as a number of cycles of the 1.024kHz CLK_WDT_OSC clock.
These bits are loaded from NVM User Row at start-up.

| Value | Name | Description |
|---|---|---|
| 0x0 | CYC8 | 8 clock cycles |
| 0x1 | CYC16 | 16 clock cycles |
| 0x2 | CYC32 | 32 clock cycles |
| 0x3 | CYC64 | 64 clock cycles |
| 0x4 | CYC128 | 128 clock cycles |
| 0x5 | CYC256 | 256 clock cycles |
| 0x6 | CYC512 | 512 clock cycles |
| 0x7 | CYC1024 | 1024 clock cycles |
| 0x8 | CYC2048 | 2048 clock cycles |
| 0x9 | CYC4096 | 4096 clock cycles |
| 0xA | CYC8192 | 8192 clock cycles |
| 0xB | CYC16384 | 16384 clock cycles |
| 0xC-0xF | Reserved | Reserved |

**Bits 3:0 – PER[3:0]** Time-Out Period
These bits determine the watchdog time-out period as a number of 1.024 kHz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period.
These bits are loaded from NVM User Row at startup.

| Value | Name | Description |
|---|---|---|
| 0x0 | CYC8 | 8 clock cycles |
| 0x1 | CYC16 | 16 clock cycles |
| 0x2 | CYC32 | 32 clock cycles |
| 0x3 | CYC64 | 64 clock cycles |
| 0x4 | CYC128 | 128 clock cycles |
| 0x5 | CYC256 | 256 clock cycles |
| 0x6 | CYC512 | 512 clock cycles |
| 0x7 | CYC1024 | 1024 clock cycles |
| 0x8 | CYC2048 | 2048 clock cycles |
| 0x9 | CYC4096 | 4096 clock cycles |
| 0xA | CYC8192 | 8192 clock cycles |
| 0xB | CYC16384 | 16384 clock cycles |
| 0xC – 0xF | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.8.3. Early Warning Control

**Name:** EWCTRL
**Offset:** 0x02
**Reset:** x initially determined from NVM User Row after reset
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | EWOFFSET[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | x | x | x | x |

**Bits 3:0 – EWOFFSET[3:0]** Early Warning Interrupt Time Offset
These bits determine the number of CLK_WDT_OSC clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from NVM User Row at start-up.

| Value | Name | Description |
|---|---|---|
| 0x0 | CYC8 | 8 clock cycles |
| 0x1 | CYC16 | 16 clock cycles |
| 0x2 | CYC32 | 32 clock cycles |
| 0x3 | CYC64 | 64 clock cycles |
| 0x4 | CYC128 | 128 clock cycles |
| 0x5 | CYC256 | 256 clock cycles |
| 0x6 | CYC512 | 512 clock cycles |
| 0x7 | CYC1024 | 1024 clock cycles |
| 0x8 | CYC2048 | 2048 clock cycles |
| 0x9 | CYC4096 | 4096 clock cycles |
| 0xA | CYC8192 | 8192 clock cycles |
| 0xB - 0xF | Reserved | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.8.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | EW |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – EW** Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Early Warning Interrupt Enable bit, which disables the Early Warning interrupt.

| Value | Description |
|---|---|
| 0 | The Early Warning interrupt is disabled. |
| 1 | The Early Warning interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.8.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | EW |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – EW** Early Warning Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit sets the Early Warning Interrupt Enable bit, which enables the Early Warning interrupt.

| Value | Description |
|---|---|
| 0 | The Early Warning interrupt is disabled. |
| 1 | The Early Warning interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.8.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x06
**Reset:** 0x00
**Property:** N/A

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | EW |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – EW** Early Warning
This flag is cleared by writing a '1' to it.
This flag is set when an Early Warning interrupt occurs, as defined by the EWOFFSET bit group in EWCTRL.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Early Warning interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 20.8.7. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | CLEAR | ALWAYSON | WEN | ENABLE | |
| Access | | | | R | R | R | R | |
| Reset | | | | 0 | 0 | 0 | 0 | |

**Bit 4 – CLEAR** Clear Synchronization Busy

| Value | Description |
|---|---|
| 0 | Write synchronization of the CLEAR register is complete. |
| 1 | Write synchronization of the CLEAR register is ongoing. |

**Bit 3 – ALWAYSON** Always-On Synchronization Busy

| Value | Description |
|---|---|
| 0 | Write synchronization of the CTRLA.ALWAYSON bit is complete. |
| 1 | Write synchronization of the CTRLA.ALWAYSON bit is ongoing. |

**Bit 2 – WEN** Window Enable Synchronization Busy

| Value | Description |
|---|---|
| 0 | Write synchronization of the CTRLA.WEN bit is complete. |
| 1 | Write synchronization of the CTRLA.WEN bit is ongoing. |

**Bit 1 – ENABLE** Enable Synchronization Busy

| Value | Description |
|---|---|
| 0 | Write synchronization of the CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization of the CTRLA.ENABLE bit is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 20.8.8. Clear

**Name:** CLEAR
**Offset:** 0x0C
**Reset:** 0x00
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CLEAR[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – CLEAR[7:0]** Watchdog Clear
In Normal mode, writing 0xA5 to this register during the watchdog time-out period will clear the Watchdog Timer and the watchdog time-out period is restarted.
In Window mode, any writing attempt to this register before the time-out period started (i.e., during $TO_{WDTW}$) will issue an immediate system Reset. Writing 0xA5 during the time-out period $TO_{WDT}$ will clear the Watchdog Timer and the complete time-out sequence (first $TO_{WDTW}$ then $TO_{WDT}$) is restarted.
In both modes, writing any other value than 0xA5 will issue an immediate system Reset.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 21. RTC – Real-Time Counter

## 21.1. Overview

The Real-Time Counter (RTC) is a 32-bit counter with a 10-bit programmable prescaler that typically runs continuously to keep track of time. The RTC can wake up the device from sleep modes using the alarm/compare wake up, periodic wake up, or overflow wake up mechanisms, or from the wake inputs.

The RTC can generate periodic peripheral events from outputs of the prescaler, as well as alarm/compare interrupts and peripheral events, which can trigger at any counter value. Additionally, the timer can trigger an overflow interrupt and overflow event, and can be reset on the occurrence of an alarm/compare match. This allows periodic interrupts and peripheral events at very long and accurate intervals.

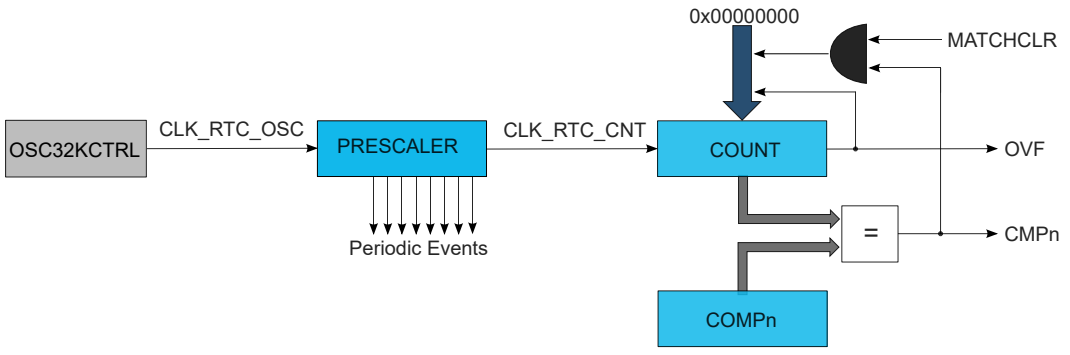
The 10-bit programmable prescaler can scale down the clock source. By this, a wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the minimum counter tick interval is 30.5µs, and time-out periods can range up to 36 hours. For a counter tick interval of 1s, the maximum time-out period is more than 136 years.

## 21.2. Features

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- Two 32-bit or four 16-bit compare values
- Clock/Calendar mode
  - Time in seconds, minutes, and hours (12/24)
  - Date in day of month, month, and year
  - Leap year correction
- Digital prescaler correction/tuning for increased accuracy
- Overflow, alarm/compare match and prescaler interrupts and events
  - Optional clear on alarm/compare match
- 8 backup registers with retention capability
- Tamper Detection
  - Timestamp on event or up to 5 inputs with debouncing
  - Active layer protection

## 21.3. Block Diagram

**Figure 21-1.** RTC Block Diagram (Mode 0 — 32-Bit Counter)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 21-2.** RTC Block Diagram (Mode 1 — 16-Bit Counter)

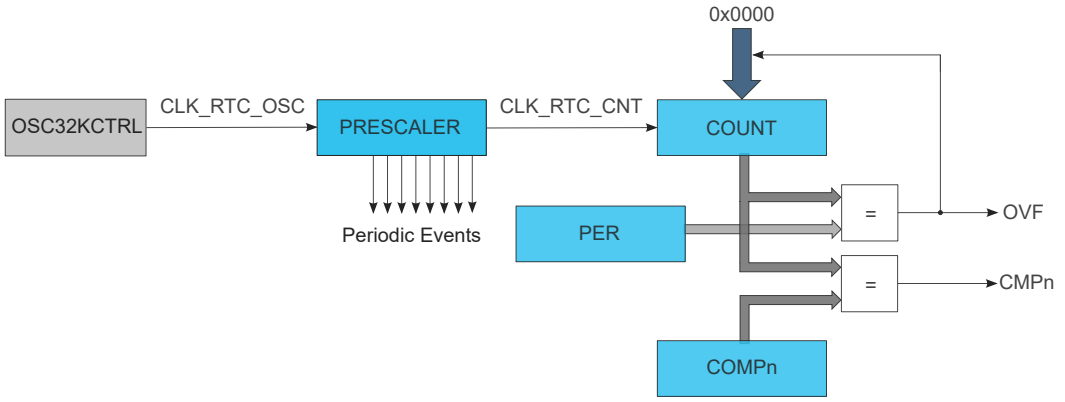
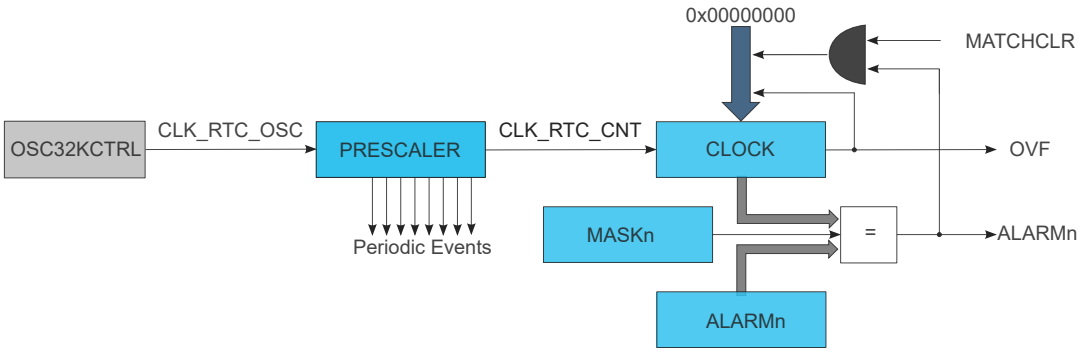



**Figure 21-3.** RTC Block Diagram (Mode 2 — Clock/Calendar)




**Related Links**
32-Bit Counter (Mode 0)
16-Bit Counter (Mode 1)
Clock/Calendar (Mode 2)
Tamper Detection

## 21.4. Signal Description

**Table 21-1.** Signal Description

| Signal | Description | Type |
|---|---|---|
| INn [n=0..4] | Tamper Detection Input | Digital input |
| OUT | Tamper Detection Output | Digital output |

One signal can be mapped to one of several pins.

**Related Links**
I/O Multiplexing and Considerations

## 21.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 21.5.1. I/O Lines

For more information on I/O configurations, refer to the "RTC Pinout" section.

**Related Links:** I/O Multiplexing and Considerations


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.5.2. Power Management

The RTC will continue to operate in any sleep modes where the selected source clock is running. The RTC interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes. Refer to the *Power Manager* for details on the different sleep modes.

The RTC can only be reset by a power on reset (POR) or by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1).

**Related Links**

PM – Power Manager

### 21.5.3. Clocks

The RTC bus clock (CLK_RTC_APB) can be enabled and disabled in the Main Clock module MCLK, and the default state of CLK_RTC_APB can be found in Peripheral Clock Masking section.

A 32.768 kHz or 1.024 kHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32.768 kHz oscillator controller (OSC32KCTRL.RTCCTRL) before using the RTC.

This oscillator clock is asynchronous to the bus clock (CLK_RTC_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**

OSC32KCTRL – 32KHz Oscillators Controller

Peripheral Clock Masking

### 21.5.4. DMA

The DMA request lines (or line if only one request) are connected to the DMA Controller (DMAC). Using the RTC DMA requests requires the DMA Controller to be configured first.

**Related Links**

DMAC – Direct Memory Access Controller

### 21.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the RTC interrupt requires the Interrupt Controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 21.5.6. Events

The events are connected to the *Event System*.

**Related Links**

Event System (EVSYS)

### 21.5.7. Debug Operation

When the CPU is halted in debug mode the RTC will halt normal operation. The RTC can be forced to continue operation during debugging. Refer to DBGCTRL for details.

### 21.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Write-protection is denoted by the "PAC Write-Protection" property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to the PAC - Peripheral Access Controller for details.

**Related Links**
PAC - Peripheral Access Controller

### 21.5.9. Analog Connections

A 32.768 kHz crystal can be connected to the XIN32 and XOUT32 pins, along with any required load capacitors. See the Electrical Characteristics Chapters for details on recommended crystal characteristics and load capacitors.

## 21.6. Functional Description

### 21.6.1. Principle of Operation

The RTC keeps track of time in the system and enables periodic events, as well as interrupts and events at a specified time. The RTC consists of a 10-bit prescaler that feeds a 32-bit counter. The actual format of the 32-bit counter depends on the RTC operating mode.

The RTC can function in one of these modes:

- Mode 0 - COUNT32: RTC serves as 32-bit counter
- Mode 1 - COUNT16: RTC serves as 16-bit counter
- Mode 2 - CLOCK: RTC serves as clock/calendar with alarm functionality

### 21.6.2. Basic Operation

#### 21.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE=0):

- Operating Mode bits in the Control A register (CTRLA.MODE)
- Prescaler bits in the Control A register (CTRLA.PRESCALER)
- Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- Clock Representation bit in the Control A register (CTRLA.CLKREP)

The following registers are enable-protected:

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- Tamper Control register (TAMPCTRL)

Enable-protected bits and registers can be changed only when the RTC is disabled (CTRLA.ENABLE=0). If the RTC is enabled (CTRLA.ENABLE=1), these operations are necessary: first write CTRLA.ENABLE=0 and check whether the write synchronization has finished, then change the desired bit field value. Enable-protected bits in CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The RTC prescaler divides the source clock for the RTC counter.

**Note:** In Clock/Calendar mode, the prescaler must be configured to provide a 1.024 kHz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

$$f_{CLK_RTC_CNT} = \frac{f_{CLK_RTC_OSC}}{2^{PRESCALER}}$$

The frequency of the oscillator clock, CLK_RTC_OSC, is given by $f_{CLK_RTC_OSC}$, and $f_{CLK_RTC_CNT}$ is the frequency of the internal prescaled RTC clock, CLK_RTC_CNT.

#### 21.6.2.2. Enabling, Disabling, and Resetting

The RTC is enabled by setting the Enable bit in the Control A register (CTRLA.ENABLE=1). The RTC is disabled by writing CTRLA.ENABLE=0.

The RTC is reset by setting the Software Reset bit in the Control A register (CTRLA.SWRST=1). All registers in the RTC, except DEBUG, will be reset to their initial state, and the RTC will be disabled. The RTC must be disabled before resetting it.

#### 21.6.2.3. 32-Bit Counter (Mode 0)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x0, the counter operates in 32-bit Counter mode. The block diagram of this mode is shown in Figure 21-1. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The counter will increment until it reaches the top value of 0xFFFFFFFF, and then wrap to 0x00000000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 32-bit format.

The counter value is continuously compared with the 32-bit Compare registers (COMPn, n=0–1). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next 0-to-1 transition of CLK_RTC_CNT.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is '1', the counter is cleared on the next counter cycle when a compare match with COMPn occurs. This allows the RTC to generate periodic interrupts or events with longer periods than the prescaler events. Note that when CTRLA.MATCHCLR is '1', INTFLAG.CMPn and INTFLAG.OVF will both be set simultaneously on a compare match with COMPn.

#### 21.6.2.4. 16-Bit Counter (Mode 1)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x1, the counter operates in 16-bit Counter mode as shown in Figure 21-2. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. In 16-bit Counter mode, the 16-bit Period register (PER) holds the maximum value of the counter. The counter will increment until it reaches the PER value, and then wrap to 0x0000. This sets the Overflow Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF).

The RTC counter value can be read from or written to the Counter Value register (COUNT) in 16-bit format.

The counter value is continuously compared with the 16-bit Compare registers (COMPn, n=0..). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..) is set on the next 0-to-1 transition of CLK_RTC_CNT.

#### 21.6.2.5. Clock/Calendar (Mode 2)

When the RTC Operating Mode bits in the Control A register (CTRLA.MODE) are written to 0x2, the counter operates in Clock/Calendar mode, as shown in Figure 21-3. When the RTC is enabled, the counter will increment on every 0-to-1 transition of CLK_RTC_CNT. The selected clock source and RTC prescaler must be configured to provide a 1Hz clock to the counter for correct operation in this mode.

The time and date can be read from or written to the Clock Value register (CLOCK) in a 32-bit time/date format. Time is represented as:

- Seconds


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Minutes
- Hours

Hours can be represented in either 12- or 24-hour format, selected by the Clock Representation bit in the Control A register (CTRLA.CLKREP). This bit can be changed only while the RTC is disabled.

The date is represented in this form:

- Day as the numeric day of the month (starting at 1)
- Month as the numeric month of the year (1 = January, 2 = February, etc.)
- Year as a value from 0x00 to 0x3F. This value must be added to a user-defined reference year. The reference year must be a leap year (2016, 2020 etc). Example: the year value 0x2D, added to a reference year 2016, represents the year 2061.

The RTC will increment until it reaches the top value of 23:59:59 December 31 of year value 0x3F, and then wrap to 00:00:00 January 1 of year value 0x00. This will set the Overflow Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.OVF).

The clock value is continuously compared with the 32-bit Alarm registers (ALARMn, n=0–1). When an alarm match occurs, the Alarm n Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARMn, n=0..1) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm 0 Interrupt flag is set with a delay of 1s after the occurrence of alarm match.

A valid alarm match depends on the setting of the Alarm Mask Selection bits in the Alarm n Mask register (MASKn.SEL). These bits determine which time/date fields of the clock and alarm values are valid for comparison and which are ignored.

If the Clear on Match bit in the Control A register (CTRLA.MATCHCLR) is set, the counter is cleared on the next counter cycle when an alarm match with ALARMn occurs. This allows the RTC to generate periodic interrupts or events with longer periods than it would be possible with the prescaler events only (see Periodic Intervals).
**Note:** When CTRLA.MATCHCLR is 1, INTFLAG.ALARM0 and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARMn.

### 21.6.3. DMA Operation

The RTC generates the following DMA request:

- Tamper (TAMPER): The request is set on capture of the timestamp. The request is cleared when the Timestamp register is read.

If the CPU accesses the registers which are source for DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

### 21.6.4. Interrupts

The RTC has the following interrupt sources:

- Overflow (OVF): Indicates that the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Indicates detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARMn): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to Periodic Intervals for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

(INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is raised and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled or the RTC is reset. See the description of the INTFLAG registers for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the Nested Vector Interrupt Controller for details. The user must read the INTFLAG register to determine which interrupt condition is present.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated. Refer to the Nested Vector Interrupt Controller for details.

**Related Links**
Nested Vector Interrupt Controller

### 21.6.5. Events

The RTC can generate the following output events:

- Overflow (OVF): Generated when the counter has reached its top value and wrapped to zero.
- Tamper (TAMPER): Generated on detection of valid signal on a tamper input pin or tamper event input.
- Compare (CMPn): Indicates a match between the counter value and the compare register.
- Alarm (ALARM): Indicates a match between the clock value and the alarm register.
- Period n (PERn): The corresponding bit in the prescaler has toggled. Refer to Periodic Intervals for details.
- Periodic Daily (PERD): Generated when the COUNT/CLOCK has incremented at a fixed period of time.

Setting the Event Output bit in the Event Control Register (EVCTRL.xxxEO=1) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the EVSYS - Event System for details on configuring the event system.

The RTC can take the following actions on an input event:

- Tamper (TAMPEVT): Capture the RTC counter to the timestamp register. See *Tamper Detection*.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.xxxEI) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event.

**Related Links**
Event System (EVSYS)

### 21.6.6. Sleep Mode Operation

The RTC will continue to operate in any sleep mode where the source clock is active. The RTC *interrupts* can be used to wake up the device from a sleep mode. RTC *events* can trigger other operations in the system without exiting the sleep mode.

An interrupt request will be generated after the wake-up if the Interrupt Controller is configured accordingly. Otherwise the CPU will wake up directly, without triggering any interrupt. In this case, the CPU will continue executing right from the first instruction that followed the entry into sleep.

The periodic events can also wake up the CPU through the interrupt function of the Event System. In this case, the event must be enabled and connected to an event channel with its interrupt enabled. See *Event System* for more information.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control A register, CTRLA.SWRST
- Enable bit in Control A register, CTRLA.ENABLE
- Count Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)
- Clock Read Synchronization bit in Control A register (CTRLA.COUNTSYNC)

The following registers are synchronized when written:

- Counter Value register, COUNT
- Clock Value register, CLOCK
- Counter Period register, PER
- Compare n Value registers, COMPn
- Alarm n Value registers, ALARMn
- Frequency Correction register, FREQCORR
- Alarm n Mask register, MASKn
- The General Purpose n registers (GPn)

The following registers are synchronized when read:

- The Counter Value register, COUNT, if the Counter Read Sync Enable bit in CTRLA (CTRLA.COUNTSYNC) is '1'
- The Clock Value register, CLOCK, if the Clock Read Sync Enable bit in CTRLA (CTRLA.CLOCKSYNC) is '1'
- The Timestamp Value register (TIMESTAMP)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

**Related Links**

Register Synchronization

### 21.6.8. Additional Features

#### 21.6.8.1. Periodic Intervals

The RTC prescaler can generate interrupts and events at periodic intervals, allowing flexible system tick creation. Any of the upper eight bits of the prescaler (bits 2 to 9) can be the source of an interrupt/event. When one of the eight Periodic Event Output bits in the Event Control register (EVCTRL.PEREO[n=0..7]) is '1', an event is generated on the 0-to-1 transition of the related bit in the prescaler, resulting in a periodic event frequency of:

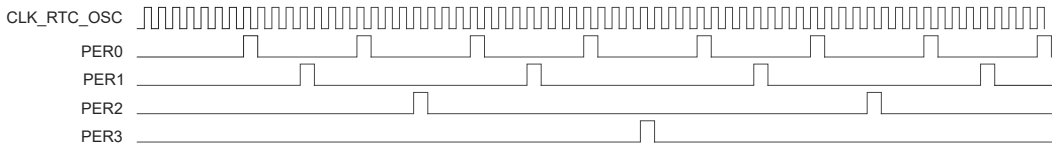
$$f_{PERIODIC(n)} = \frac{f_{CLK_RTC_OSC}}{2^{n+3}}$$

$f_{CLK_RTC_OSC}$ is the frequency of the internal prescaler clock CLK_RTC_OSC, and n is the position of the EVCTRL.PEREOn bit. For example, PER0 will generate an event every eight CLK_RTC_OSC cycles, PER1 every 16 cycles, etc. This is shown in the figure below.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Periodic events are independent of the prescaler setting used by the RTC counter, except if CTRLA.PRESCALER is zero. Then, no periodic events will be generated.

**Figure 21-4.** Example Periodic Events




**Note:** This example also applies to interrupts. Just replace EVCTRL.PEREOn with the PERn fields of INTENCLR, INTENSET, and INTFLAG. For Modes 0 and 2, n = 0,..7. For Mode 1 n = 2...7.

### 21.6.8.2. Frequency Correction

The RTC Frequency Correction module employs periodic counter corrections to compensate for a too-slow or too-fast oscillator. Frequency correction requires that CTRLA.PRESCALER is greater than 1.

The digital correction circuit adds or subtracts cycles from the RTC prescaler to adjust the frequency in approximately 1ppm steps. Digital correction is achieved by adding or skipping a single count in the prescaler once every 8192 CLK_RTC_OSC cycles. The Value bit group in the Frequency Correction register (FREQCORR.VALUE) determines the number of times the adjustment is applied over 128 of these periods. The resulting correction is as follows:

$$\text{Correction in ppm} = \frac{\text{FREQCORR.VALUE}}{8192 \cdot 128} \cdot 10^6 \text{ppm}$$

This results in a resolution of 0.95367ppm.

The Sign bit in the Frequency Correction register (FREQCORR.SIGN) determines the direction of the correction. A positive value will add counts and increase the period (reducing the frequency), and a negative value will reduce counts per period (speeding up the frequency).

Digital correction also affects the generation of the periodic events from the prescaler. When the correction is applied at the end of the correction cycle period, the interval between the previous periodic event and the next occurrence may also be shortened or lengthened depending on the correction value.

### 21.6.8.3. Backup Registers

The RTC includes eight Backup registers (BKUPn). These registers maintain their content in Backup sleep mode. They can be used to store user-defined values.

If more user-defined data must be stored than the eight Backup registers can hold, the General Purpose registers (GPn) can be used.

**Related Links**

PM – Power Manager

### 21.6.8.4. General Purpose Registers

The RTC includes four General Purpose registers (GPn). These registers are reset only when the RTC is reset or when tamper detection occurs while CTRLA.GPTRST=1, and remain powered while the RTC is powered. They can be used to store user-defined values while other parts of the system are powered off.

The general purpose registers 2*n and 2*n+1 are enabled by writing a '1' to the General Purpose Enable bit n in the Control B register (CTRLB.GPnEN).

The GP registers share internal resources with the COMPARE/ALARM features. Each COMPARE/ALARM register have a separate read buffer and write buffer. When the general purpose feature is enabled the even GP uses the read buffer while the odd GP uses the write buffer.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When the COMPARE/ALARM register is written, the write buffer hold temporarily the COMPARE/ALARM value until the synchronisation is complete (bit SYNCBUSY.COMPn going to 0). After the write is completed the write buffer can be used as a odd general purpose register whithout affecting the COMPARE/ALARM function.

If the COMPARE/ALARM function is not used, the read buffer can be used as an even general purpose register. In this case writing the even GP will temporarirely use the write buffer until the synchronisation is complete (bit SYNCBUSY.GPn going to 0). Thus an even GP must be written before writing the odd GP. Changing or writing an even GP needs to temporarily save the value of the odd GP.

Before using an even GP, the associated COMPARE/ALARM feature must be disabled by writing a '1' to the General Purpose Enable bit in the Control B register (CTRLB.GPnEN). To re-enable the compare/alarm, CTRLB.GPnEN must be written to zero and the associated COMPn/ALARMn must be written with the correct value.

An example procedure to write the general purpose registers GP0 and GP1 is:

1. Wait for any ongoing write to COMP0 to complete (SYNCBUSY.COMP0 = 0). If the RTC is operating in Mode 1, wait for any ongoing write to COMP1 to complete as well (SYNCBUSY.COMP1 = 0).
2. Write CTRLB.GP0EN = 1 if GP0 is needed.
3. Write GP0 if needed.
4. Wait for any ongoing write to GP0 to complete (SYNCBUSY.GP0 = 0). Note that GP1 will also show as busy when GP0 is busy.
5. Write GP1 if needed.

The following table provides the correspondence of General Purpose Registers and the COMPARE/ALARM read or write buffer in all RTC modes.

**Table 21-2.** General Purpose Registers Versus Compare/Alarm Registers: n in 0, 2, 4, 6...

| Register | Mode 0 | Mode 1 | Mode 2 | Write Before |
|---|---|---|---|---|
| GPn | COMPn/2 write buffer | (COMPn , COMPn+1) write buffer | ALARMn/2 write buffer | GPn+1 |
| GPn+1 | COMPn/2 read buffer | (COMPn , COMPn+1) read buffer | ALARMn/2 read buffer | - |

### 21.6.8.5. Tamper Detection

The RTC provides five tamper channels that can be used for tamper detection.

The action of each tamper channel is configured using the Input n Action bits in the Tamper Control register (TAMPCTRL.INnACT):

- Off: Detection for tamper channel n is disabled.
- Wake: A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will not be captured in the TIMESTAMP register.
- Capture: A transition on INn input (tamper channel n) matching TAMPCTRL.TAMPLVLn will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.
- Active Layer Protection: A mismatch of an internal RTC signal routed between INn and OUTn pins will be detected and the tamper interrupt flag (INTFLAG.TAMPER) will be set. The RTC value will be captured in the TIMESTAMP register.

In order to determine which tamper source caused a tamper event, the Tamper ID register (TAMPID) provides the detection status of each tamper channel. These bits remain active until cleared by software.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

A single interrupt request (TAMPER) is available for all tamper channels.

The RTC also supports an input event (TAMPEVT) for generating a tamper condition within the Event System. The tamper input event is enabled by the Tamper Input Event Enable bit in the Event Control register (EVCTRL.TAMPEVTEI).

Up to five polarity external inputs (INn) can be used for tamper detection. The polarity for each input is selected with the Tamper Level bits in the Tamper Control register (TAMPCTRL.TAMPLVLn).

Separate debouncers are embedded for each external input. The debouncer for each input is enabled/disabled with the Debounce Enable bits in the Tamper Control register (TAMPCTRL.DEBNCn). The debouncer configuration is fixed for all inputs as set by the Control B register (CTRLB). The debouncing period duration is configurable using the Debounce Frequency field in the Control B register (CTRLB.DEBF). The period is set for all debouncers (i.e., the duration cannot be adjusted separately for each debouncer).

When TAMPCTRL.DEBNCn = 0, INn is detected asynchronously. See Figure 21-5 for an example.

When TAMPCTRL.DEBNCn = 1, the detection time depends on whether the debouncer operates synchronously or asynchronously, and whether majority detection is enabled or not. Refer to the table below for more details. Synchronous versus asynchronous stability debouncing is configured by the Debounce Asynchronous Enable bit in the Control B register (CTRLB.DEBASYNC):

- Synchronous (CTRLB.DEBASYNC = 0): INn is synchronized in two CLK_RTC periods and then must remain stable for four CLK_RTC_DEB periods before a valid detection occurs. See Figure 21-6 for an example.
- Asynchronous (CTRLB.DEBASYNC = 1): The first edge on INn is detected. Further detection is blanked until INn remains stable for four CLK_RTC_DEB periods. See Figure 21-7 for an example.

Majority debouncing is configured by the Debounce Majority Enable bit in the Control B register (CTRLB.DEBMAJ). INn must be valid for two out of three CLK_RTC_DEB periods. See Figure 21-8 for an example.

**Table 21-3.** Debouncer Configuration

| TAMPCTRL. DEBNCn | CTRLB. DEBMAJ | CTRLB. DEBASYNC | Description |
|---|---|---|---|
| 0 | X | X | Detect edge on INn with no debouncing. Every edge detected is immediately triggered. |
| 1 | 0 | 0 | Detect edge on INn with synchronous stability debouncing. Edge detected is only triggered when INn is stable for 4 consecutive CLK_RTC_DEB periods. |
| 1 | 0 | 1 | Detect edge on INn with asynchronous stability debouncing. First detected edge is triggered immediately. All subsequent detected edges are ignored until INn is stable for 4 consecutive CLK_RTC_DEB periods. |
| 1 | 1 | X | Detect edge on INn with majority debouncing. Pin INn is sampled for 3 consecutive CLK_RTC_DEB periods. Signal level is determined by majority-rule (LLL, LLH, LHL, HLL = '0' and LHH, HLH, HHL, HHH = '1'). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 21-5.** Edge Detection with Debouncer Disabled

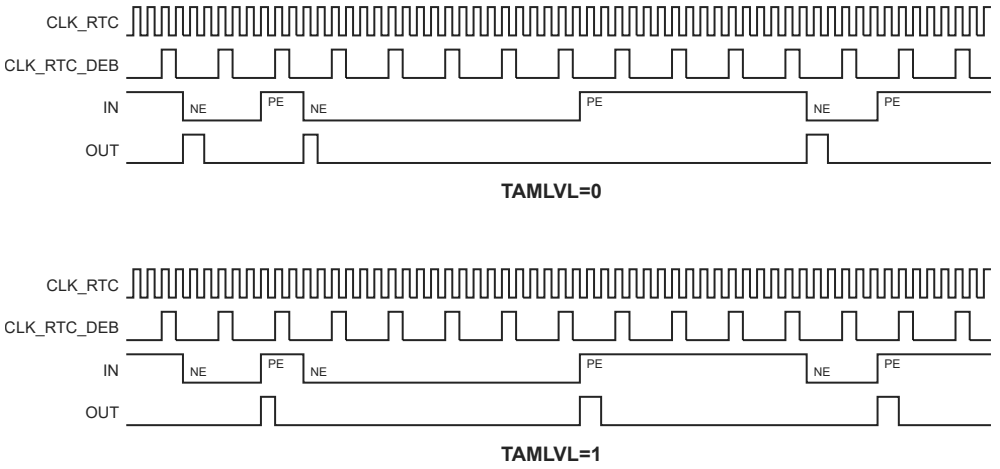
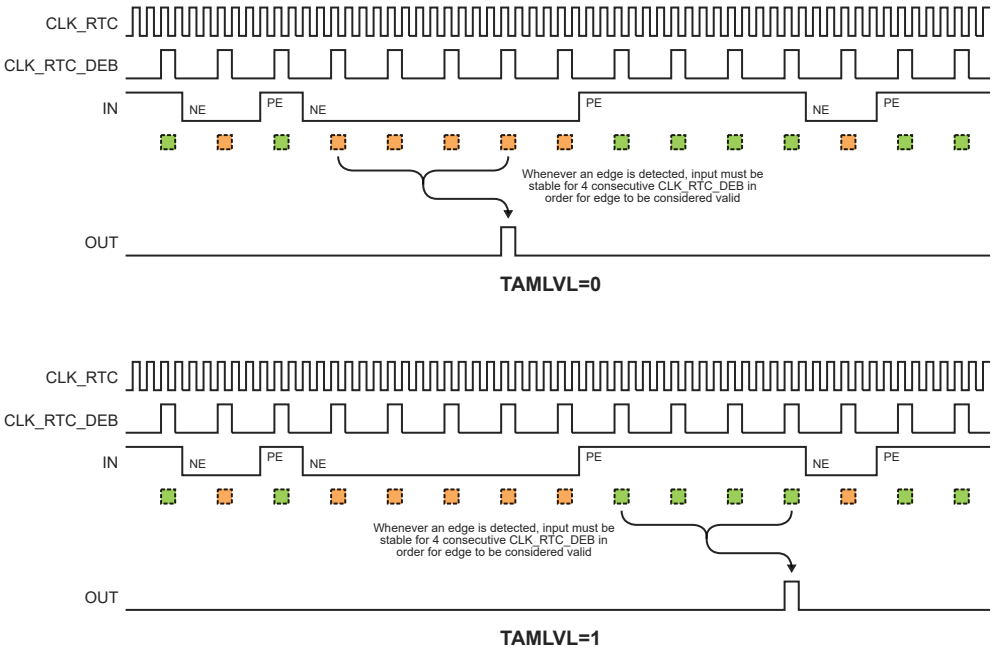



**Figure 21-6.** Edge Detection with Synchronous Stability Debouncing





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 21-7.** Edge Detection with Asynchronous Stability Debouncing

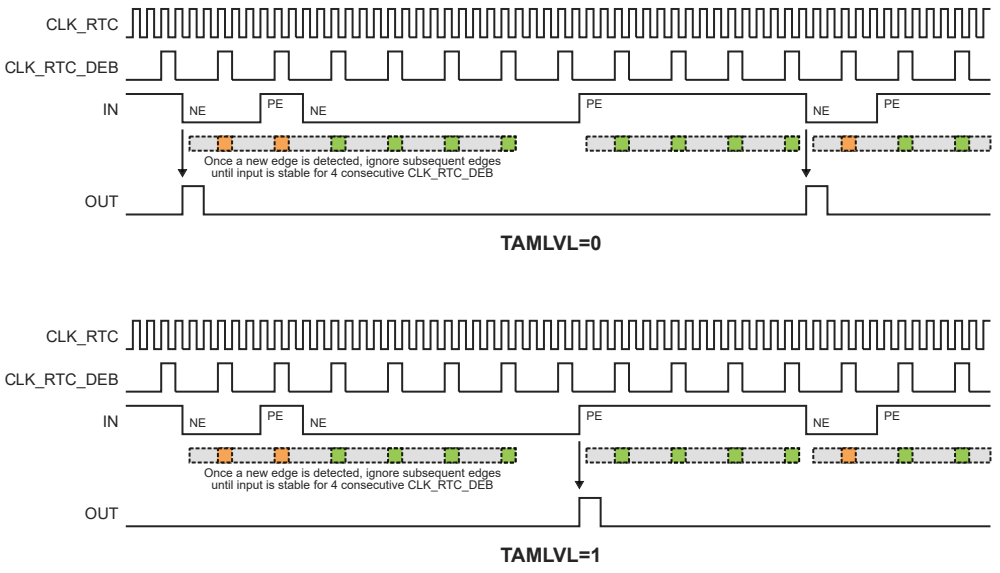
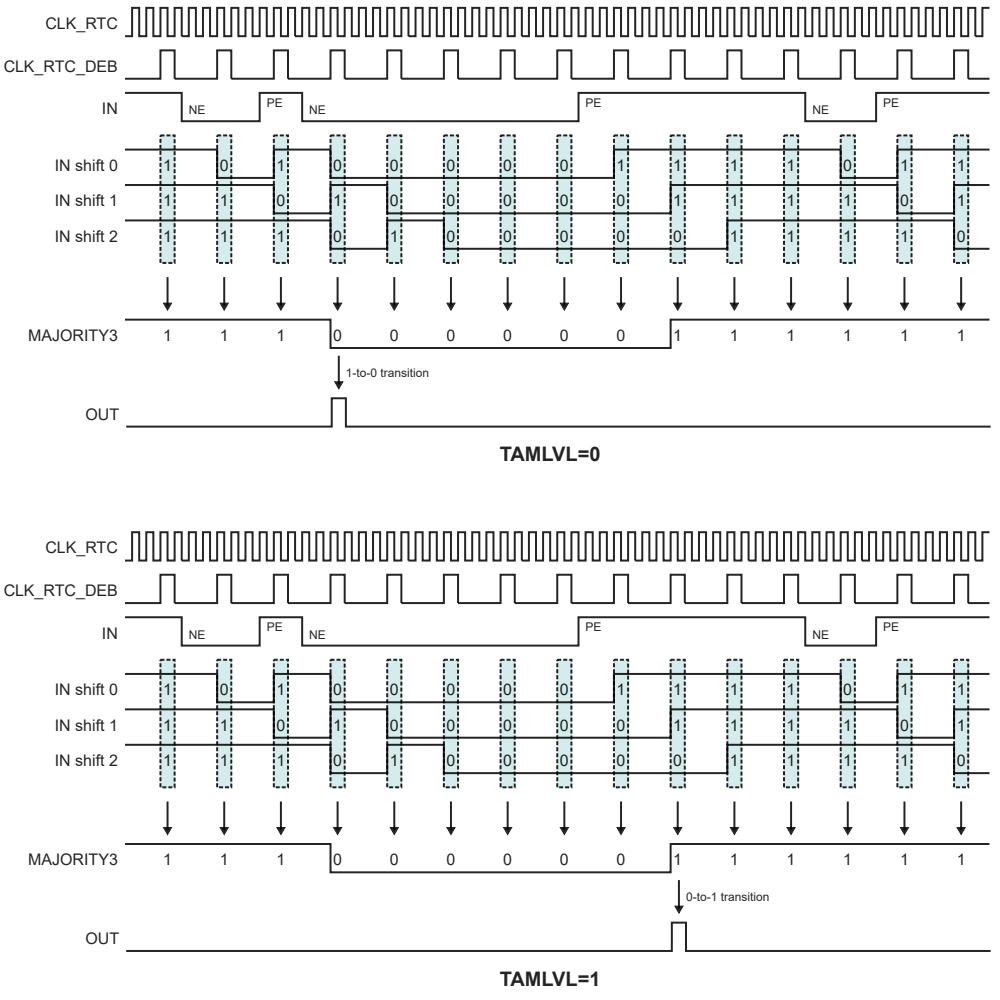



**Figure 21-8.** Edge Detection with Majority Debouncing





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Block Diagram
Timestamp
Active Layer Protection

#### 21.6.8.5.1. Timestamp

As part of tamper detection the RTC can capture the counter value (COUNT/CLOCK) into the TIMESTAMP register. Three CLK_RTC periods are required to detect the tampering condition and capture the value. The TIMESTAMP value can be read once the Tamper flag in the Interrupt Flag register (INTFLAG.TAMPER) is set. If the DMA Enable bit in the Control B register (CTRLB.DMAEN) is '1', a DMA request will be triggered by the timestamp. In order to determine which tamper source caused a capture, the Tamper ID register (TAMPID) provides the detection status of each tamper channel and the tamper input event. A DMA transfer can then read both TIMESTAMP and TAMPID in succession.

A new timestamp value cannot be captured until the Tamper flag is cleared, either by reading the timestamp or by writing a '1' to INTFLAG.TAMPER. If several tamper conditions occur in a short window before the flag is cleared, only the first timestamp may be logged. However, the detection of each tamper will still be recorded in TAMPID.

The Tamper Input Event (TAMPEVT) will always perform a timestamp capture. To capture on the external inputs (INn), the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT) must be written to '1'. If an input is set for wake functionality it does not capture the timestamp; however the Tamper flag and TAMPID will still be updated.
**Note:** The TIMESTAMP value should be read once, and INTFLAG.TAMPER must be cleared. The next value should be read only after the INTFLAG.TAMPER is set again.

**Related Links**
Tamper Detection

#### 21.6.8.5.2. Active Layer Protection

The RTC provides a mean of detecting broken traces on the PCB , also known as Active layer Protection. In this mode, a generated internal RTC signal can be directly routed over critical components on the board using RTC OUT output pin to one of the RTC INn input pins. A tamper condition is detected if there is a mismatch on the generated RTC signal.

The Active Layer Protection mode and the generation of the RTC signal is enabled by setting the RTCOUT bit in the Control B register (CTRLB.RTCOUT).

Enabling active layer protection requires the following steps:

- Enable the RTC prescaler output by writing a one to the RTC Out bit in the Control B register (CTRLB.RTCOUT). The I/O pins must also be configured to correctly route the signal to the external pins.
- Select the frequency of the output signal by configuring the RTC Active Layer Frequency field in the Control B register (CTRLB.ACTF).

$$\text{GCLK_RTC_OUT} = \frac{\text{CLK_RTC}}{2^{\text{CTRLB.ACTF}+1}}$$

- Enable the tamper input n (INn) in active layer mode by writing 3 to the corresponding Input Action field in the Tamper Control register (TAMPCTRL.INnACT). When active layer protection is enabled and INn and OUTn pin are used, the value of INn is sampled on the falling edge of CLK_RTC and compared to the expected value of OUTn. Therefore up to one half of a CLK_RTC period is available for propagation delay through the trace.
- Enable Active Layer Protection by setting CTRLB.RTCOUT bit.

**Related Links**
Tamper Detection


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 21.7. Register Summary - Mode 0 - 32-Bit Counter

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | MATCHCLR | | | | MODE[1:0] | | ENABLE | SWRST |
| | | 15:8 | COUNTSYNC | GPTRST | BKTRST | | PRESCALER[3:0] | | | |
| 0x02 | CTRLB | 7:0 | DMAEN | RTCOUT | DEBASYNC | DEBMAJ | | | GP2EN | GP0EN |
| | | 15:8 | | ACTF[2:0] | | | | DEBF[2:0] | | |
| 0x04 | EVCTRL | 7:0 | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| | | 15:8 | OVFEO | TAMPEREO | | | | | CMPEO1 | CMPEO0 |
| | | 23:16 | | | | | | | | TAMPEVEI |
| | | 31:24 | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | | | CMP1 | CMP0 |
| 0x0A | INTENSET | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | | | CMP1 | CMP0 |
| 0x0C | INTFLAG | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | | | CMP1 | CMP0 |
| 0x0E | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0F | Reserved | | | | | | | | | |
| 0x10 | SYNCBUSY | 7:0 | | COMP1 | COMP0 | | COUNT | FREQCORR | ENABLE | SWRST |
| | | 15:8 | COUNTSYNC | | | | | | | |
| | | 23:16 | | | | | GP3 | GP2 | GP1 | GP0 |
| | | 31:24 | | | | | | | | |
| 0x14 | FREQCORR | 7:0 | SIGN | VALUE[6:0] | | | | | | |
| 0x15 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | COUNT[23:16] | | | | | | | |
| | | 31:24 | COUNT[31:24] | | | | | | | |
| 0x1C ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | COMP0 | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| | | 23:16 | COMP[23:16] | | | | | | | |
| | | 31:24 | COMP[31:24] | | | | | | | |
| 0x24 | COMP1 | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| | | 23:16 | COMP[23:16] | | | | | | | |
| | | 31:24 | COMP[31:24] | | | | | | | |
| 0x28 ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | GP0 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x44 | GP1 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x48 | GP2 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x4C | GP3 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary - Mode 0 - 32-Bit Counter** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x50 ... 0x5F | Reserved | | | | | | | | | |
| 0x60 | TAMPCTRL | 7:0 | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| | | 15:8 | | | | | | | IN4ACT[1:0] | |
| | | 23:16 | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| | | 31:24 | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| 0x64 | TIMESTAMP | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | COUNT[23:16] | | | | | | | |
| | | 31:24 | COUNT[31:24] | | | | | | | |
| 0x68 | TAMPID | 7:0 | | | | TAMPID4 | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TAMPEVT | | | | | | | |
| 0x6C ... 0x7F | Reserved | | | | | | | | | |
| 0x80 | BKUP0 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x84 | BKUP1 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x88 | BKUP2 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x8C | BKUP3 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x90 | BKUP4 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x94 | BKUP5 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x98 | BKUP6 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x9C | BKUP7 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |

## 21.8. Register Description - Mode 0 - 32-Bit Counter

This Register Description section is valid if the RTC is in COUNT32 mode (CTRLA.MODE=0).

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 21.8.1. Control A in COUNT32 mode (CTRLA.MODE=0)

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNTSYNC | GPTRST | BKTRST | | PRESCALER[3:0] | | | |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MATCHCLR | | | | MODE[1:0] | | ENABLE | SWRST |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 15 – COUNTSYNC** COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | COUNT read synchronization is disabled |
| 1 | COUNT read synchronization is enabled |

**Bit 14 – GPTRST** GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

**Bit 13 – BKTRST** BKUP Registers Reset On Tamper Enable
All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | BKUPn registers will not reset when a tamper condition occurs. |
| 1 | BKUPn registers will reset when a tamper condition occurs. |

**Bits 11:8 – PRESCALER[3:0]** Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | CLK_RTC_CNT = GCLK_RTC/1 |
| 0x1 | DIV1 | CLK_RTC_CNT = GCLK_RTC/1 |
| 0x2 | DIV2 | CLK_RTC_CNT = GCLK_RTC/2 |
| 0x3 | DIV4 | CLK_RTC_CNT = GCLK_RTC/4 |
| 0x4 | DIV8 | CLK_RTC_CNT = GCLK_RTC/8 |
| 0x5 | DIV16 | CLK_RTC_CNT = GCLK_RTC/16 |
| 0x6 | DIV32 | CLK_RTC_CNT = GCLK_RTC/32 |
| 0x7 | DIV64 | CLK_RTC_CNT = GCLK_RTC/64 |
| 0x8 | DIV128 | CLK_RTC_CNT = GCLK_RTC/128 |
| 0x9 | DIV256 | CLK_RTC_CNT = GCLK_RTC/256 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0xA | DIV512 | CLK_RTC_CNT = GCLK_RTC/512 |
| 0xB | DIV1024 | CLK_RTC_CNT = GCLK_RTC/1024 |
| 0xC-0xF | - | Reserved |

**Bit 7 – MATCHCLR** Clear on Match

This bit defines if the counter is cleared or not on a match.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The counter is not cleared on a Compare/Alarm 0 match |
| 1 | The counter is cleared on a Compare/Alarm 0 match |

**Bits 3:2 – MODE[1:0]** Operating Mode

This bit group defines the operating mode of the RTC.
This bit is not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | COUNT32 | Mode 0: 32-bit counter |
| 0x1 | COUNT16 | Mode 1: 16-bit counter |
| 0x2 | CLOCK | Mode 2: Clock/calendar |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE** Enable

Due to synchronization there is a delay between writing CTRLA.ENABLE and until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled |
| 1 | The peripheral is enabled |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay between writing CTRLA.SWRST and until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

| Value | Description |
|---|---|
| 0 | There is not reset operation ongoing |
| 1 | The reset operation is ongoing |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.2. Control B in COUNT32 mode (CTRLA.MODE=0)

**Name:** CTRLB
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | ACTF[2:0] | | | | DEBF[2:0] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DMAEN | RTCOUT | DEBASYNC | DEBMAJ | | | GP2EN | GP0EN |
| Access | R/W | R/W | R/W | R/W | | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | | 0 | 0 |

**Bits 14:12 – ACTF[2:0]** Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | CLK_RTC_OUT = CLK_RTC / 2 |
| 0x1 | DIV4 | CLK_RTC_OUT = CLK_RTC / 4 |
| 0x2 | DIV8 | CLK_RTC_OUT = CLK_RTC / 8 |
| 0x3 | DIV16 | CLK_RTC_OUT = CLK_RTC / 16 |
| 0x4 | DIV32 | CLK_RTC_OUT = CLK_RTC / 32 |
| 0x5 | DIV64 | CLK_RTC_OUT = CLK_RTC / 64 |
| 0x6 | DIV128 | CLK_RTC_OUT = CLK_RTC / 128 |
| 0x7 | DIV256 | CLK_RTC_OUT = CLK_RTC / 256 |

**Bits 10:8 – DEBF[2:0]** Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | CLK_RTC_DEB = CLK_RTC / 2 |
| 0x1 | DIV4 | CLK_RTC_DEB = CLK_RTC / 4 |
| 0x2 | DIV8 | CLK_RTC_DEB = CLK_RTC / 8 |
| 0x3 | DIV16 | CLK_RTC_DEB = CLK_RTC / 16 |
| 0x4 | DIV32 | CLK_RTC_DEB = CLK_RTC / 32 |
| 0x5 | DIV64 | CLK_RTC_DEB = CLK_RTC / 64 |
| 0x6 | DIV128 | CLK_RTC_DEB = CLK_RTC / 128 |
| 0x7 | DIV256 | CLK_RTC_DEB = CLK_RTC / 256 |

**Bit 7 – DMAEN** DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

| Value | Description |
|---|---|
| 0 | Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER. |
| 1 | Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER. |

**Bit 6 – RTCOUT** RTC Output Enable

| Value | Description |
|---|---|
| 0 | The RTC active layer output is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The RTC active layer output is enabled. |

**Bit 5 – DEBASYNC** Debouncer Asynchronous Enable

| Value | Description |
|---|---|
| 0 | The tamper input debouncers operate synchronously. |
| 1 | The tamper input debouncers operate asynchronously. |

**Bit 4 – DEBMAJ** Debouncer Majority Enable

| Value | Description |
|---|---|
| 0 | The tamper input debouncers match three equal values. |
| 1 | The tamper input debouncers match majority two of three values. |

**Bit 1 – GP2EN** General Purpose 2 Enable

| Value | Description |
|---|---|
| 0 | COMP1 compare function enabled. GP2/GP3 disabled. |
| 1 | COMP1 compare function disabled. GP2/GP3 enabled. |

**Bit 0 – GP0EN** General Purpose 0 Enable

| Value | Description |
|---|---|
| 0 | COMP0 compare function enabled. GP0/GP1 disabled. |
| 1 | COMP0 compare function disabled. GP0/GP1 enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.3. Event Control in COUNT32 mode (CTRLA.MODE=0)

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | TAMPEVEI |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVFEO | TAMPEREO | | | | | CMPEO1 | CMPEO0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 16 – TAMPEVEI** Tamper Event Input Enable

| Value | Description |
|---|---|
| 0 | Tamper event input is disabled and incoming events will be ignored. |
| 1 | Tamper event input is enabled and incoming events will capture the COUNT value. |

**Bit 15 – OVFEO** Overflow Event Output Enable

| Value | Description |
|---|---|
| 0 | Overflow event is disabled and will not be generated. |
| 1 | Overflow event is enabled and will be generated for every overflow. |

**Bit 14 – TAMPEREO** Tamper Event Output Enable

| Value | Description |
|---|---|
| 0 | Tamper event output is disabled and will not be generated. |
| 1 | Tamper event output is enabled and will be generated for every tamper input. |

**Bits 8, 9 – CMPEOn** Compare n Event Output Enable [n = 1..0]

| Value | Description |
|---|---|
| 0 | Compare n event is disabled and will not be generated. |
| 1 | Compare n event is enabled and will be generated for every compare match. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn** Periodic Interval n Event Output Enable [n = 7..0]

| Value | Description |
|---|---|
| 0 | Periodic Interval n event is disabled and will not be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Periodic Interval n event is enabled and will be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 21.8.4. Interrupt Enable Clear in COUNT32 mode (CTRLA.MODE=0)

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | | | CMP1 | CMP0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**Bit 14 – TAMPER** Tamper Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this but will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

| Value | Description |
|---|---|
| 0 | The Tamper interrupt is disabled. |
| 1 | The Tamper interrupt is enabled. |

**Bits 8, 9 – CMPn** Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

| Value | Description |
|---|---|
| 0 | The Compare n interrupt is disabled |
| 1 | The Compare n interrupt is enabled. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

| Value | Description |
|---|---|
| 0 | Periodic Interval n interrupt is disabled. |
| 1 | Periodic Interval n interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.5. Interrupt Enable Set in COUNT32 mode (CTRLA.MODE=0)

**Name:** INTENSET
**Offset:** 0x0A
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | | | CMP1 | CMP0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**Bit 14 – TAMPER** Tamper Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

| Value | Description |
|---|---|
| 0 | The Tamper interrupt is disabled. |
| 1 | The Tamper interrupt is enabled. |

**Bits 8, 9 – CMPn** Compare n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which and enables the Compare n interrupt.

| Value | Description |
|---|---|
| 0 | The Compare n interrupt is disabled. |
| 1 | The Compare n interrupt is enabled. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

| Value | Description |
|---|---|
| 0 | Periodic Interval n interrupt is disabled. |
| 1 | Periodic Interval n interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.6. Interrupt Flag Status and Clear in COUNT32 mode (CTRLA.MODE=0)

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | | | CMP1 | CMP0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

**Bit 14 – TAMPER** Tamper event
This flag is set after a damper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

**Bits 8, 9 – CMPn** Compare n [n = 1..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMPn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.7. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0E
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The RTC is halted when the CPU is halted by an external debugger. |
| 1 | The RTC continues normal operation when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 21.8.8. Synchronization Busy in COUNT32 mode (CTRLA.MODE=0)

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | GP3 | GP2 | GP1 | GP0 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNTSYNC | | | | | | | |
| Access | R | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | COMP1 | COMP0 | | COUNT | FREQCORR | ENABLE | SWRST |
| Access | | R | R | | R | R | R | R |
| Reset | | 0 | 0 | | 0 | 0 | 0 | 0 |

**Bits 16, 17, 18, 19 – GPn** General Purpose n Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for GPn register is complete. |
| 1 | Write synchronization for GPn register is ongoing. |

**Bit 15 – COUNTSYNC** Count Read Sync Enable Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.COUNTSYNC bit is complete. |
| 1 | Write synchronization for CTRLA.COUNTSYNC bit is ongoing. |

**Bits 5, 6 – COMPn** Compare n Synchronization Busy Status [n = 1..0]

| Value | Description |
|---|---|
| 0 | Write synchronization for COMPx register is complete. |
| 1 | Write synchronization for COMPx register is ongoing. |

**Bit 3 – COUNT** Count Value Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Read/write synchronization for COUNT register is complete. |
| 1 | Read/write synchronization for COUNT register is ongoing. |

**Bit 2 – FREQCORR** Frequency Correction Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for FREQCORR register is complete. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Write synchronization for FREQCORR register is ongoing. |

**Bit 1 – ENABLE** Enable Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization for CTRLA.ENABLE bit is ongoing. |

**Bit 0 – SWRST** Software Reset Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.SWRST bit is complete. |
| 1 | Write synchronization for CTRLA.SWRST bit is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 21.8.9. Frequency Correction

**Name:** FREQCORR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SIGN | VALUE[6:0] | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – SIGN** Correction Sign

| Value | Description |
|---|---|
| 0 | The correction value is positive, i.e., frequency will be decreased. |
| 1 | The correction value is negative, i.e., frequency will be increased. |

**Bits 6:0 – VALUE[6:0]** Correction Value
These bits define the amount of correction applied to the RTC prescaler.

| Value | Description |
|---|---|
| 0 | Correction is disabled and the RTC frequency is unchanged. |
| 1 - 127 | The RTC frequency is adjusted according to the value. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.10. Counter Value in COUNT32 mode (CTRLA.MODE=0)

**Name:** COUNT
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

**Notes:**

1. This register is read-synchronized when CTRLA.COUNTSYNC = 1: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.
2. This register is write-synchronized: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – COUNT[31:0]** Counter Value
These bits define the value of the 32-bit RTC counter in mode 0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.11. Compare n Value in COUNT32 mode (CTRLA.MODE=0)

**Name:** COMPn
**Offset:** 0x20 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | COMP[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | COMP[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COMP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COMP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – COMP[31:0]** Compare Value

The 32-bit value of COMPn is continuously compared with the 32-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle, and the counter value is cleared if CTRLA.MATCHCLR is one.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.12. General Purpose n

**Name:** GPn
**Offset:** 0x40 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | GP[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | GP[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | GP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – GP[31:0]** General Purpose
These bits are for user-defined general purpose use, see General Purpose Registers.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.13. Tamper Control

**Name:** TAMPCTRL
**Offset:** 0x60
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | IN4ACT[1:0] | |
| Access | | | | | | | | |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 24, 25, 26, 27, 28 – DEBNC** Debounce Enable of Tamper Input INn
**Note:** Debounce feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

| Value | Description |
|---|---|
| 0 | Debouncing is disabled for Tamper input INn |
| 1 | Debouncing is enabled for Tamper input INn |

**Bits 16, 17, 18, 19, 20 – TAMLVL** Tamper Level Select of Tamper Input INn
**Note:** Tamper Level feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

| Value | Description |
|---|---|
| 0 | A falling edge condition will be detected on Tamper input INn. |
| 1 | A rising edge condition will be detected on Tamper input INn. |

**Bits 0:1, 2:3, 4:5, 6:7, 8:9 – INACT** Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Off (Disabled) |
| 0x1 | WAKE | Wake and set Tamper flag |
| 0x2 | CAPTURE | Capture timestamp and set Tamper flag |
| 0x3 | ACTL | Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.14. Timestamp

**Name:** TIMESTAMP
**Offset:** 0x64
**Reset:** 0x0
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[31:24] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[23:16] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | RO | RO | RO | RO | RO | RO | RO | RO |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – COUNT[31:0]** Count Timestamp Value
The 32-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.15. Tamper ID

**Name:** TAMPID
**Offset:** 0x68
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TAMPEVT | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | TAMPID4 | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – TAMPEVT** Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

| Value | Description |
|---|---|
| 0 | A tamper input event has not been detected |
| 1 | A tamper input event has been detected |

**Bits 0, 1, 2, 3, 4 – TAMPID** Tamper on Channel n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

| Value | Description |
|---|---|
| 0 | A tamper condition has not been detected on Channel n |
| 1 | A tamper condition has been detected on Channel n |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.8.16. Backup n

**Name:** BKUP
**Offset:** 0x80 + n*0x04 [n=0..7]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – BKUP[31:0]** Backup
These bits are user-defined for general purpose use in the Backup domain.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 21.9. Register Summary - Mode 1 - 16-Bit Counter

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | MODE[1:0] | | ENABLE | SWRST |
| | | 15:8 | COUNTSYNC | GPTRST | BKTRST | | PRESCALER[3:0] | | | |
| 0x02 | CTRLB | 7:0 | DMAEN | RTCOUT | DEBASYNC | DEBMAJ | | | GP2EN | GP0EN |
| | | 15:8 | | ACTF[2:0] | | | | DEBF[2:0] | | |
| 0x04 | EVCTRL | 7:0 | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| | | 15:8 | OVFEO | TAMPEREO | | | CMPEO3 | CMPEO2 | CMPEO1 | CMPEO0 |
| | | 23:16 | | | | | | | | TAMPEVEI |
| | | 31:24 | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | CMP3 | CMP2 | CMP1 | CMP0 |
| 0x0A | INTENSET | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | CMP3 | CMP2 | CMP1 | CMP0 |
| 0x0C | INTFLAG | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | CMP3 | CMP2 | CMP1 | CMP0 |
| 0x0E | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0F | Reserved | | | | | | | | | |
| 0x10 | SYNCBUSY | 7:0 | COMP2 | COMP1 | COMP0 | PER | COUNT | FREQCORR | ENABLE | SWRST |
| | | 15:8 | COUNTSYNC | | | | | | | COMP3 |
| | | 23:16 | | | | | GP3 | GP2 | GP1 | GP0 |
| | | 31:24 | | | | | | | | |
| 0x14 | FREQCORR | 7:0 | SIGN | VALUE[6:0] | | | | | | |
| 0x15 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| 0x1A ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | PER | 7:0 | PER[7:0] | | | | | | | |
| | | 15:8 | PER[15:8] | | | | | | | |
| 0x1E ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | COMP0 | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| 0x22 | COMP1 | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| 0x24 | COMP2 | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| 0x26 | COMP3 | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| 0x28 ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | GP0 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x44 | GP1 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x48 | GP2 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary - Mode 1 - 16-Bit Counter** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x4C | GP3 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x50 ... 0x5F | Reserved | | | | | | | | | |
| 0x60 | TAMPCTRL | 7:0 | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| | | 15:8 | | | | | | | IN4ACT[1:0] | |
| | | 23:16 | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| | | 31:24 | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| 0x64 | TIMESTAMP | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x68 | TAMPID | 7:0 | | | | TAMPID4 | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TAMPEVT | | | | | | | |
| 0x6C ... 0x7F | Reserved | | | | | | | | | |
| 0x80 | BKUP0 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x84 | BKUP1 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x88 | BKUP2 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x8C | BKUP3 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x90 | BKUP4 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x94 | BKUP5 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x98 | BKUP6 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x9C | BKUP7 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |

## 21.10. Register Description - Mode 1 - 16-Bit Counter

This Register Description section is valid if the RTC is in COUNT16 mode (CTRLA.MODE=1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 21.10.1. Control A in COUNT16 mode (CTRLA.MODE=1)

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNTSYNC | GPTRST | BKTRST | | PRESCALER[3:0] | | | |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | MODE[1:0] | | ENABLE | SWRST |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 15 – COUNTSYNC** COUNT Read Synchronization Enable
The COUNT register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the COUNT register.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | COUNT read synchronization is disabled |
| 1 | COUNT read synchronization is enabled |

**Bit 14 – GPTRST** GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | GPn registers will not reset when a tamper condition occurs. |
| 1 | GPn registers will reset when a tamper condition occurs. |

**Bit 13 – BKTRST** BKUP Registers Reset On Tamper Enable
All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | BKUPn registers will not reset when a tamper condition occurs. |
| 1 | BKUPn registers will reset when a tamper condition occurs. |

**Bits 11:8 – PRESCALER[3:0]** Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | CLK_RTC_CNT = GCLK_RTC/1 |
| 0x1 | DIV1 | CLK_RTC_CNT = GCLK_RTC/1 |
| 0x2 | DIV2 | CLK_RTC_CNT = GCLK_RTC/2 |
| 0x3 | DIV4 | CLK_RTC_CNT = GCLK_RTC/4 |
| 0x4 | DIV8 | CLK_RTC_CNT = GCLK_RTC/8 |
| 0x5 | DIV16 | CLK_RTC_CNT = GCLK_RTC/16 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x6 | DIV32 | CLK_RTC_CNT = GCLK_RTC/32 |
| 0x7 | DIV64 | CLK_RTC_CNT = GCLK_RTC/64 |
| 0x8 | DIV128 | CLK_RTC_CNT = GCLK_RTC/128 |
| 0x9 | DIV256 | CLK_RTC_CNT = GCLK_RTC/256 |
| 0xA | DIV512 | CLK_RTC_CNT = GCLK_RTC/512 |
| 0xB | DIV1024 | CLK_RTC_CNT = GCLK_RTC/1024 |
| 0xC-0xF | - | Reserved |

**Bits 3:2 – MODE[1:0]** Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | COUNT32 | Mode 0: 32-bit counter |
| 0x1 | COUNT16 | Mode 1: 16-bit counter |
| 0x2 | CLOCK | Mode 2: Clock/calendar |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE** Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled |
| 1 | The peripheral is enabled |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC (except DBGCTRL) to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

| Value | Description |
|---|---|
| 0 | There is not reset operation ongoing |
| 1 | The reset operation is ongoing |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.2. Control B in COUNT16 mode (CTRLA.MODE=1)

**Name:** CTRLB
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | ACTF[2:0] | | | | DEBF[2:0] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DMAEN | RTCOUT | DEBASYNC | DEBMAJ | | | GP2EN | GP0EN |
| Access | R/W | R/W | R/W | R/W | | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | | 0 | 0 |

**Bits 14:12 – ACTF[2:0]** Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | CLK_RTC_OUT = CLK_RTC / 2 |
| 0x1 | DIV4 | CLK_RTC_OUT = CLK_RTC / 4 |
| 0x2 | DIV8 | CLK_RTC_OUT = CLK_RTC / 8 |
| 0x3 | DIV16 | CLK_RTC_OUT = CLK_RTC / 16 |
| 0x4 | DIV32 | CLK_RTC_OUT = CLK_RTC / 32 |
| 0x5 | DIV64 | CLK_RTC_OUT = CLK_RTC / 64 |
| 0x6 | DIV128 | CLK_RTC_OUT = CLK_RTC / 128 |
| 0x7 | DIV256 | CLK_RTC_OUT = CLK_RTC / 256 |

**Bits 10:8 – DEBF[2:0]** Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | CLK_RTC_DEB = CLK_RTC / 2 |
| 0x1 | DIV4 | CLK_RTC_DEB = CLK_RTC / 4 |
| 0x2 | DIV8 | CLK_RTC_DEB = CLK_RTC / 8 |
| 0x3 | DIV16 | CLK_RTC_DEB = CLK_RTC / 16 |
| 0x4 | DIV32 | CLK_RTC_DEB = CLK_RTC / 32 |
| 0x5 | DIV64 | CLK_RTC_DEB = CLK_RTC / 64 |
| 0x6 | DIV128 | CLK_RTC_DEB = CLK_RTC / 128 |
| 0x7 | DIV256 | CLK_RTC_DEB = CLK_RTC / 256 |

**Bit 7 – DMAEN** DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

| Value | Description |
|---|---|
| 0 | Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER. |
| 1 | Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER. |

**Bit 6 – RTCOUT** RTC Output Enable

| Value | Description |
|---|---|
| 0 | The RTC active layer output is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The RTC active layer output is enabled. |

**Bit 5 – DEBASYNC** Debouncer Asynchronous Enable

| Value | Description |
|---|---|
| 0 | The tamper input debouncers operate synchronously. |
| 1 | The tamper input debouncers operate asynchronously. |

**Bit 4 – DEBMAJ** Debouncer Majority Enable

| Value | Description |
|---|---|
| 0 | The tamper input debouncers match three equal values. |
| 1 | The tamper input debouncers match majority two of three values. |

**Bit 1 – GP2EN** General Purpose 2 Enable

| Value | Description |
|---|---|
| 0 | COMP1 compare function enabled. GP2/GP3 disabled. |
| 1 | COMP1 compare function disabled. GP2/GP3 enabled. |

**Bit 0 – GP0EN** General Purpose 0 Enable

| Value | Description |
|---|---|
| 0 | COMP0 compare function enabled. GP0/GP1 disabled. |
| 1 | COMP0 compare function disabled. GP0/GP1 enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.3. Event Control in COUNT16 mode (CTRLA.MODE=1)

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | TAMPEVEI |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVFEO | TAMPEREO | | | CMPEO3 | CMPEO2 | CMPEO1 | CMPEO0 |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 16 – TAMPEVEI** Tamper Event Input Enable

| Value | Description |
|---|---|
| 0 | Tamper event input is disabled, and incoming events will be ignored |
| 1 | Tamper event input is enabled, and incoming events will capture the COUNT value |

**Bit 15 – OVFEO** Overflow Event Output Enable

| Value | Description |
|---|---|
| 0 | Overflow event is disabled and will not be generated. |
| 1 | Overflow event is enabled and will be generated for every overflow. |

**Bit 14 – TAMPEREO** Tamper Event Output Enable

| Value | Description |
|---|---|
| 0 | Tamper event output is disabled, and will not be generated. |
| 1 | Tamper event output is enabled, and will be generated for every tamper input. |

**Bits 8, 9, 10, 11 – CMPEOn** Compare n Event Output Enable [n = 3..0]

| Value | Description |
|---|---|
| 0 | Compare n event is disabled and will not be generated. |
| 1 | Compare n event is enabled and will be generated for every compare match. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn** Periodic Interval n Event Output Enable [n = 7..0]

| Value | Description |
|---|---|
| 0 | Periodic Interval n event is disabled and will not be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Periodic Interval n event is enabled and will be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 21.10.4. Interrupt Enable Clear in COUNT16 mode (CTRLA.MODE=1)

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | CMP3 | CMP2 | CMP1 | CMP0 |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**Bit 14 – TAMPER** Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Tamper Interrupt Enable bit, which disables the Tamper interrupt.

| Value | Description |
|---|---|
| 0 | The Tamper interrupt is disabled. |
| 1 | The Tamper interrupt is enabled. |

**Bits 8, 9, 10, 11 – CMPn** Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Compare n Interrupt Enable bit, which disables the Compare n interrupt.

| Value | Description |
|---|---|
| 0 | The Compare n interrupt is disabled. |
| 1 | The Compare n interrupt is enabled. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

| Value | Description |
|---|---|
| 0 | Periodic Interval n interrupt is disabled. |
| 1 | Periodic Interval n interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.5. Interrupt Enable Set in COUNT16 mode (CTRLA.MODE=1)

**Name:** INTENSET
**Offset:** 0x0A
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | CMP3 | CMP2 | CMP1 | CMP0 |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**Bit 14 – TAMPER** Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

| Value | Description |
|---|---|
| 0 | The Tamper interrupt is disabled. |
| 1 | The Tamper interrupt is enabled. |

**Bits 8, 9, 10, 11 – CMPn** Compare n Interrupt Enable [n = 3..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Compare n Interrupt Enable bit, which and enables the Compare n interrupt.

| Value | Description |
|---|---|
| 0 | The Compare n interrupt is disabled. |
| 1 | The Compare n interrupt is enabled. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

| Value | Description |
|---|---|
| 0 | Periodic Interval n interrupt is disabled. |
| 1 | Periodic Interval n interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.6. Interrupt Flag Status and Clear in COUNT16 mode (CTRLA.MODE=1)

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | CMP3 | CMP2 | CMP1 | CMP0 |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

**Bit 14 – TAMPER** Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/ INTENSET.TAMPER is one.
Writing a '0' to this bit has no effect.
Writing a one to this bit clears the Tamper interrupt flag.

**Bits 8, 9, 10, 11 – CMPn** Compare n [n = 3..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.COMPn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Compare n interrupt flag.

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.7. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0E
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The RTC is halted when the CPU is halted by an external debugger. |
| 1 | The RTC continues normal operation when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 21.10.8. Synchronization Busy in COUNT16 mode (CTRLA.MODE=1)

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | GP3 | GP2 | GP1 | GP0 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNTSYNC | | | | | | | COMP3 |
| Access | R | | | | | | | R |
| Reset | 0 | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COMP2 | COMP1 | COMP0 | PER | COUNT | FREQCORR | ENABLE | SWRST |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 16, 17, 18, 19 – GPn** General Purpose n Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for GPn register is complete. |
| 1 | Write synchronization for GPn register is ongoing. |

**Bit 15 – COUNTSYNC** Count Read Sync Enable Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.COUNTSYNC bit is complete. |
| 1 | Write synchronization for CTRLA.COUNTSYNC bit is ongoing. |

**Bits 5, 6, 7, 8 – COMPn** Compare n Synchronization Busy Status [n = 3..0]

| Value | Description |
|---|---|
| 0 | Write synchronization for COMPn register is complete. |
| 1 | Write synchronization for COMPn register is ongoing. |

**Bit 4 – PER** Period Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for PER register is complete. |
| 1 | Write synchronization for PER register is ongoing. |

**Bit 3 – COUNT** Count Value Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Read/write synchronization for COUNT register is complete. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Read/write synchronization for COUNT register is ongoing. |

**Bit 2 – FREQCORR** Frequency Correction Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for FREQCORR register is complete. |
| 1 | Write synchronization for FREQCORR register is ongoing. |

**Bit 1 – ENABLE** Enable Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization for CTRLA.ENABLE bit is ongoing. |

**Bit 0 – SWRST** Software Reset Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.SWRST bit is complete. |
| 1 | Write synchronization for CTRLA.SWRST bit is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 21.10.9. Frequency Correction

**Name:** FREQCORR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SIGN | VALUE[6:0] | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – SIGN** Correction Sign

| Value | Description |
|---|---|
| 0 | The correction value is positive, i.e., frequency will be decreased. |
| 1 | The correction value is negative, i.e., frequency will be increased. |

**Bits 6:0 – VALUE[6:0]** Correction Value
These bits define the amount of correction applied to the RTC prescaler.

| Value | Description |
|---|---|
| 0 | Correction is disabled and the RTC frequency is unchanged. |
| 1 - 127 | The RTC frequency is adjusted according to the value. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.10. Counter Value in COUNT16 mode (CTRLA.MODE=1)

**Name:** COUNT
**Offset:** 0x18
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

**Notes:**

1. This register is read-synchronized when CTRLA.COUNTSYNC = 1: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.
2. This register is write-synchronized: SYNCBUSY.COUNT must be checked to ensure the COUNT register synchronization is complete.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – COUNT[15:0]** Counter Value
These bits define the value of the 16-bit RTC counter in COUNT16 mode (CTRLA.MODE=1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.11. Counter Period in COUNT16 mode (CTRLA.MODE=1)

**Name:** PER
**Offset:** 0x1C
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PER[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – PER[15:0]** Counter Period
These bits define the value of the 16-bit RTC period in COUNT16 mode (CTRLA.MODE=1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.12. Compare n Value in COUNT16 mode (CTRLA.MODE=1)

**Name:** COMP
**Offset:** 0x20 + n*0x02 [n=0..3]
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COMP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COMP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – COMP[15:0]** Compare Value

The 16-bit value of COMPn is continuously compared with the 16-bit COUNT value. When a match occurs, the Compare n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn) is set on the next counter cycle.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.13. General Purpose n

**Name:** GPn
**Offset:** 0x40 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | GP[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | GP[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | GP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – GP[31:0]** General Purpose
These bits are for user-defined general purpose use, see General Purpose Registers.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.14. Tamper Control

**Name:** TAMPCTRL
**Offset:** 0x60
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | IN4ACT[1:0] | |
| Access | | | | | | | | |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 24, 25, 26, 27, 28 – DEBNC** Debounce Enable of Tamper Input INn
**Note:** Debounce feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

| Value | Description |
|---|---|
| 0 | Debouncing is disabled for Tamper input INn |
| 1 | Debouncing is enabled for Tamper input INn |

**Bits 16, 17, 18, 19, 20 – TAMLVL** Tamper Level Select of Tamper Input INn
**Note:** Tamper Level feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

| Value | Description |
|---|---|
| 0 | A falling edge condition will be detected on Tamper input INn. |
| 1 | A rising edge condition will be detected on Tamper input INn. |

**Bits 0:1, 2:3, 4:5, 6:7, 8:9 – INACT** Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Off (Disabled) |
| 0x1 | WAKE | Wake and set Tamper flag |
| 0x2 | CAPTURE | Capture timestamp and set Tamper flag |
| 0x3 | ACTL | Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.15. Timestamp

**Name:** TIMESTAMP
**Offset:** 0x64
**Reset:** 0x0000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – COUNT[15:0]** Count Timestamp Value
The 16-bit value of COUNT is captured by the TIMESTAMP when a tamper condition occurs.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.16. Tamper ID

**Name:** TAMPID
**Offset:** 0x68
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TAMPEVT | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | TAMPID4 | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – TAMPEVT** Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

| Value | Description |
|---|---|
| 0 | A tamper input event has not been detected |
| 1 | A tamper input event has been detected |

**Bits 0, 1, 2, 3, 4 – TAMPID** Tamper on Channel n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

| Value | Description |
|---|---|
| 0 | A tamper condition has not been detected on Channel n |
| 1 | A tamper condition has been detected on Channel n |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.10.17. Backup n

**Name:** BKUP
**Offset:** 0x80 + n*0x04 [n=0..7]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – BKUP[31:0]** Backup
These bits are user-defined for general purpose use in the Backup domain.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 21.11. Register Summary - Mode 2 - Clock/Calendar

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | MATCHCLR | CLKREP | | | MODE[1:0] | | ENABLE | SWRST |
| | | 15:8 | CLOCKSYNC | GPTRST | BKTRST | | PRESCALER[3:0] | | | |
| 0x02 | CTRLB | 7:0 | DMAEN | RTCOUT | DEBASYNC | DEBMAJ | | | GP2EN | GP0EN |
| | | 15:8 | | ACTF[2:0] | | | | DEBF[2:0] | | |
| 0x04 | EVCTRL | 7:0 | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| | | 15:8 | OVFEO | TAMPEREO | | | | | ALARMEO1 | ALARMEO0 |
| | | 23:16 | | | | | | | | TAMPEVEI |
| | | 31:24 | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | | | ALARM1 | ALARM0 |
| 0x0A | INTENSET | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | | | ALARM1 | ALARM0 |
| 0x0C | INTFLAG | 7:0 | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| | | 15:8 | OVF | TAMPER | | | | | ALARM1 | ALARM0 |
| 0x0E | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0F | Reserved | | | | | | | | | |
| 0x10 | SYNCBUSY | 7:0 | | ALARM1 | ALARM0 | | CLOCK | FREQCORR | ENABLE | SWRST |
| | | 15:8 | CLOCKSYNC | | | MASK1 | MASK0 | | | |
| | | 23:16 | | | | | GP3 | GP2 | GP1 | GP0 |
| | | 31:24 | | | | | | | | |
| 0x14 | FREQCORR | 7:0 | SIGN | VALUE[6:0] | | | | | | |
| 0x15 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | CLOCK | 7:0 | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| | | 15:8 | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| | | 23:16 | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| | | 31:24 | YEAR[5:0] | | | | | | MONTH[3:2] | |
| 0x1C ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | ALARM0 | 7:0 | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| | | 15:8 | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| | | 23:16 | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| | | 31:24 | YEAR[5:0] | | | | | | MONTH[3:2] | |
| 0x24 | MASK0 | 7:0 | | | | | | SEL[2:0] | | |
| 0x25 ... 0x27 | Reserved | | | | | | | | | |
| 0x28 | ALARM1 | 7:0 | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| | | 15:8 | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| | | 23:16 | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| | | 31:24 | YEAR[5:0] | | | | | | MONTH[3:2] | |
| 0x2C | MASK1 | 7:0 | | | | | | SEL[2:0] | | |
| 0x2D ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | GP0 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x44 | GP1 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x48 | GP2 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary - Mode 2 - Clock/Calendar** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x4C | GP3 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x50 ... 0x5F | Reserved | | | | | | | | | |
| 0x60 | TAMPCTRL | 7:0 | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| | | 15:8 | | | | | | | IN4ACT[1:0] | |
| | | 23:16 | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| | | 31:24 | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| 0x64 | TIMESTAMP | 7:0 | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| | | 15:8 | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| | | 23:16 | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| | | 31:24 | YEAR[5:0] | | | | | | MONTH[3:2] | |
| 0x68 | TAMPID | 7:0 | | | | TAMPID4 | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TAMPEVT | | | | | | | |
| 0x6C ... 0x7F | Reserved | | | | | | | | | |
| 0x80 | BKUP0 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x84 | BKUP1 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x88 | BKUP2 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x8C | BKUP3 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x90 | BKUP4 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x94 | BKUP5 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x98 | BKUP6 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |
| 0x9C | BKUP7 | 7:0 | BKUP[7:0] | | | | | | | |
| | | 15:8 | BKUP[15:8] | | | | | | | |
| | | 23:16 | BKUP[23:16] | | | | | | | |
| | | 31:24 | BKUP[31:24] | | | | | | | |

## 21.12. Register Description - Mode 2 - Clock/Calendar

This Register Description section is valid if the RTC is in Clock/Calendar mode (CTRLA.MODE=2).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 21.12.1. Control A in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CLOCKSYNC | GPTRST | BKTRST | | PRESCALER[3:0] | | | |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MATCHCLR | CLKREP | | | MODE[1:0] | | ENABLE | SWRST |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

**Bit 15 – CLOCKSYNC** CLOCK Read Synchronization Enable
The CLOCK register requires synchronization when reading. Disabling the synchronization will prevent reading valid values from the CLOCK register.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | CLOCK read synchronization is disabled |
| 1 | CLOCK read synchronization is enabled |

**Bit 14 – GPTRST** GP Registers Reset On Tamper Enable
Only GP registers enabled by the CTRLB.GPnEN bits are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

**Bit 13 – BKTRST** BKUP Registers Reset On Tamper Enable
All BKUPn registers are affected. This bit can be written only when the peripheral is disabled.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | BKUPn registers will not reset when a tamper condition occurs. |
| 1 | BKUPn registers will reset when a tamper condition occurs. |

**Bits 11:8 – PRESCALER[3:0]** Prescaler
These bits define the prescaling factor for the RTC clock source (GCLK_RTC) to generate the counter clock (CLK_RTC_CNT). Periodic events and interrupts are not available when the prescaler is off.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | CLK_RTC_CNT = GCLK_RTC/1 |
| 0x1 | DIV1 | CLK_RTC_CNT = GCLK_RTC/1 |
| 0x2 | DIV2 | CLK_RTC_CNT = GCLK_RTC/2 |
| 0x3 | DIV4 | CLK_RTC_CNT = GCLK_RTC/4 |
| 0x4 | DIV8 | CLK_RTC_CNT = GCLK_RTC/8 |
| 0x5 | DIV16 | CLK_RTC_CNT = GCLK_RTC/16 |
| 0x6 | DIV32 | CLK_RTC_CNT = GCLK_RTC/32 |
| 0x7 | DIV64 | CLK_RTC_CNT = GCLK_RTC/64 |
| 0x8 | DIV128 | CLK_RTC_CNT = GCLK_RTC/128 |
| 0x9 | DIV256 | CLK_RTC_CNT = GCLK_RTC/256 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0xA | DIV512 | CLK_RTC_CNT = GCLK_RTC/512 |
| 0xB | DIV1024 | CLK_RTC_CNT = GCLK_RTC/1024 |
| 0xC-0xF | - | Reserved |

**Bit 7 – MATCHCLR** Clear on Match

This bit is valid only in Mode 0 (COUNT32) and Mode 2 (CLOCK). This bit can be written only when the peripheral is disabled. This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The counter is not cleared on a Compare/Alarm 0 match |
| 1 | The counter is cleared on a Compare/Alarm 0 match |

**Bit 6 – CLKREP** Clock Representation

This bit is valid only in Mode 2 and determines how the hours are represented in the Clock Value (CLOCK) register. This bit can be written only when the peripheral is disabled. This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | 24 Hour |
| 1 | 12 Hour (AM/PM) |

**Bits 3:2 – MODE[1:0]** Operating Mode

This field defines the operating mode of the RTC. This bit is not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | COUNT32 | Mode 0: 32-bit counter |
| 0x1 | COUNT16 | Mode 1: 16-bit counter |
| 0x2 | CLOCK | Mode 2: Clock/calendar |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE** Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled |
| 1 | The peripheral is enabled |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the RTC, except DBGCTRL, to their initial state, and the RTC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST will be cleared when the reset is complete.

| Value | Description |
|---|---|
| 0 | There is not reset operation ongoing |
| 1 | The reset operation is ongoing |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.2. Control B in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** CTRLB
**Offset:** 0x2
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | ACTF[2:0] | | | | DEBF[2:0] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DMAEN | RTCOUT | DEBASYNC | DEBMAJ | | | GP2EN | GP0EN |
| Access | R/W | R/W | R/W | R/W | | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | | 0 | 0 |

**Bits 14:12 – ACTF[2:0]** Active Layer Frequency
These bits define the prescaling factor for the RTC clock output (OUT) used during active layer protection in terms of the CLK_RTC.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | CLK_RTC_OUT = CLK_RTC / 2 |
| 0x1 | DIV4 | CLK_RTC_OUT = CLK_RTC / 4 |
| 0x2 | DIV8 | CLK_RTC_OUT = CLK_RTC / 8 |
| 0x3 | DIV16 | CLK_RTC_OUT = CLK_RTC / 16 |
| 0x4 | DIV32 | CLK_RTC_OUT = CLK_RTC / 32 |
| 0x5 | DIV64 | CLK_RTC_OUT = CLK_RTC / 64 |
| 0x6 | DIV128 | CLK_RTC_OUT = CLK_RTC / 128 |
| 0x7 | DIV256 | CLK_RTC_OUT = CLK_RTC / 256 |

**Bits 10:8 – DEBF[2:0]** Debounce Frequency
These bits define the prescaling factor for the input debouncers in terms of the CLK_RTC.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | CLK_RTC_DEB = CLK_RTC / 2 |
| 0x1 | DIV4 | CLK_RTC_DEB = CLK_RTC / 4 |
| 0x2 | DIV8 | CLK_RTC_DEB = CLK_RTC / 8 |
| 0x3 | DIV16 | CLK_RTC_DEB = CLK_RTC / 16 |
| 0x4 | DIV32 | CLK_RTC_DEB = CLK_RTC / 32 |
| 0x5 | DIV64 | CLK_RTC_DEB = CLK_RTC / 64 |
| 0x6 | DIV128 | CLK_RTC_DEB = CLK_RTC / 128 |
| 0x7 | DIV256 | CLK_RTC_DEB = CLK_RTC / 256 |

**Bit 7 – DMAEN** DMA Enable
The RTC can trigger a DMA request when the timestamp is ready in the TIMESTAMP register.

| Value | Description |
|---|---|
| 0 | Tamper DMA request is disabled. Reading TIMESTAMP has no effect on INTFLAG.TAMPER. |
| 1 | Tamper DMA request is enabled. Reading TIMESTAMP will clear INTFLAG.TAMPER. |

**Bit 6 – RTCOUT** RTC Out Enable

| Value | Description |
|---|---|
| 0 | The RTC active layer output is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The RTC active layer output is enabled. |

**Bit 5 – DEBASYNC** Debouncer Asynchronous Enable

| Value | Description |
|---|---|
| 0 | The tamper input debouncers operate synchronously. |
| 1 | The tamper input debouncers operate asynchronously. |

**Bit 4 – DEBMAJ** Debouncer Majority Enable

| Value | Description |
|---|---|
| 0 | The tamper input debouncers match three equal values. |
| 1 | The tamper input debouncers match majority two of three values. |

**Bit 1 – GP2EN** General Purpose 2 Enable

| Value | Description |
|---|---|
| 0 | COMP1 compare function enabled. GP2/GP3 disabled. |
| 1 | COMP1 compare function disabled. GP2/GP3 enabled. |

**Bit 0 – GP0EN** General Purpose 0 Enable

| Value | Description |
|---|---|
| 0 | COMP0 compare function enabled. GP0 disabled. |
| 1 | COMP0 compare function disabled. GP0 enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.3. Event Control in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | TAMPEVEI |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVFEO | TAMPEREO | | | | | ALARMEO1 | ALARMEO0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 16 – TAMPEVEI** Tamper Event Input Enable

| Value | Description |
|---|---|
| 0 | Tamper event input is disabled, and incoming events will be ignored. |
| 1 | Tamper event input is enabled, and all incoming events will capture the CLOCK value. |

**Bit 15 – OVFEO** Overflow Event Output Enable

| Value | Description |
|---|---|
| 0 | Overflow event is disabled and will not be generated. |
| 1 | Overflow event is enabled and will be generated for every overflow. |

**Bit 14 – TAMPEREO** Tamper Event Output Enable

| Value | Description |
|---|---|
| 0 | Tamper event output is disabled, and will not be generated |
| 1 | Tamper event output is enabled, and will be generated for every tamper input. |

**Bits 8, 9 – ALARMEOn** Alarm n Event Output Enable [n = 1..0]

| Value | Description |
|---|---|
| 0 | Alarm n event is disabled and will not be generated. |
| 1 | Alarm n event is enabled and will be generated for every compare match. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PEREOn** Periodic Interval n Event Output Enable [n = 7..0]

| Value | Description |
|---|---|
| 0 | Periodic Interval n event is disabled and will not be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Periodic Interval n event is enabled and will be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 21.12.4. Interrupt Enable Clear in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | | | ALARM1 | ALARM0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**Bit 14 – TAMPER** Tamper Interrupt Enable

**Bits 8, 9 – ALARMn** Alarm n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Alarm n Interrupt Enable bit, which disables the Alarm n interrupt.

| Value | Description |
|---|---|
| 0 | The Alarm n interrupt is disabled. |
| 1 | The Alarm n interrupt is enabled. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Periodic Interval n Interrupt Enable bit, which disables the Periodic Interval n interrupt.

| Value | Description |
|---|---|
| 0 | Periodic Interval n interrupt is disabled. |
| 1 | Periodic Interval n interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.5. Interrupt Enable Set in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** INTENSET
**Offset:** 0x0A
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | | | ALARM1 | ALARM0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**Bit 14 – TAMPER** Tamper Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Tamper Interrupt Enable bit, which enables the Tamper interrupt.

| Value | Description |
|---|---|
| 0 | The Tamper interrupt it disabled. |
| 1 | The Tamper interrupt is enabled. |

**Bits 8, 9 – ALARMn** Alarm n Interrupt Enable [n = 1..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Alarm n Interrupt Enable bit, which and enables the Alarm n interrupt.

| Value | Description |
|---|---|
| 0 | The Alarm n interrupt is disabled. |
| 1 | The Alarm n interrupt is enabled. |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n Interrupt Enable [n = 7..0]
Writing a '0' to this bit has no effect. Writing a '1' to this bit will set the Periodic Interval n Interrupt Enable bit, which enables the Periodic Interval n interrupt.

| Value | Description |
|---|---|
| 0 | Periodic Interval n interrupt is disabled. |
| 1 | Periodic Interval n interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.6. Interrupt Flag Status and Clear in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | | | ALARM1 | ALARM0 |
| Access | R/W | R/W | | | | | R/W | R/W |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER7 | PER6 | PER5 | PER4 | PER3 | PER2 | PER1 | PER0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – OVF** Overflow
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after an overflow condition occurs, and an interrupt request will be generated if INTENCLR/SET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.

**Bit 14 – TAMPER** Tamper
This flag is set after a tamper condition occurs, and an interrupt request will be generated if INTENCLR.TAMPER/INTENSET.TAMPER is '1'. Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the Tamper interrupt flag.

**Bits 8, 9 – ALARMn** Alarm n [n = 1..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the next CLK_RTC_CNT cycle after a match with the compare condition, and an interrupt request will be generated if INTENCLR/SET.ALARMn is one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Alarm n interrupt flag.

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PERn** Periodic Interval n [n = 7..0]
This flag is cleared by writing a '1' to the flag.
This flag is set on the 0-to-1 transition of prescaler bit [n+2], and an interrupt request will be generated if INTENCLR/SET.PERx is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Periodic Interval n interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.7. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0E
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The RTC is halted when the CPU is halted by an external debugger. |
| 1 | The RTC continues normal operation when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.8. Synchronization Busy in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | GP3 | GP2 | GP1 | GP0 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CLOCKSYNC | | | MASK1 | MASK0 | | | |
| Access | R | | | R | R | | | |
| Reset | 0 | | | 0 | 0 | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | ALARM1 | ALARM0 | | CLOCK | FREQCORR | ENABLE | SWRST |
| Access | | R | R | | R | R | R | R |
| Reset | | 0 | 0 | | 0 | 0 | 0 | 0 |

**Bits 16, 17, 18, 19 – GPn** General Purpose n Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for GPn register is complete. |
| 1 | Write synchronization for GPn register is ongoing. |

**Bit 15 – CLOCKSYNC** Clock Read Sync Enable Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.CLOCKSYNC bit is complete. |
| 1 | Write synchronization for CTRLA.CLOCKSYNC bit is ongoing. |

**Bits 11, 12 – MASKn** Mask n Synchronization Busy Status [n = 1..0]

| Value | Description |
|---|---|
| 0 | Write synchronization for MASKx register is complete. |
| 1 | Write synchronization for MASKx register is ongoing. |

**Bits 5, 6 – ALARMn** Alarm n Synchronization Busy Status [n = 1..0]

| Value | Description |
|---|---|
| 0 | Write synchronization for ALARMx register is complete. |
| 1 | Write synchronization for ALARMx register is ongoing. |

**Bit 3 – CLOCK** Clock Register Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Read/write synchronization for CLOCK register is complete. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Read/write synchronization for CLOCK register is ongoing. |

**Bit 2 – FREQCORR** Frequency Correction Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for FREQCORR register is complete. |
| 1 | Write synchronization for FREQCORR register is ongoing. |

**Bit 1 – ENABLE** Enable Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization for CTRLA.ENABLE bit is ongoing. |

**Bit 0 – SWRST** Software Reset Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.SWRST bit is complete. |
| 1 | Write synchronization for CTRLA.SWRST bit is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.9. Frequency Correction

**Name:** FREQCORR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SIGN | VALUE[6:0] | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – SIGN** Correction Sign

| Value | Description |
|---|---|
| 0 | The correction value is positive, i.e., frequency will be decreased. |
| 1 | The correction value is negative, i.e., frequency will be increased. |

**Bits 6:0 – VALUE[6:0]** Correction Value
These bits define the amount of correction applied to the RTC prescaler.

| Value | Description |
|---|---|
| 0 | Correction is disabled and the RTC frequency is unchanged. |
| 1 - 127 | The RTC frequency is adjusted according to the value. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.10. Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** CLOCK
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | YEAR[5:0] | | | | | | MONTH[3:2] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:26 – YEAR[5:0]** Year
The year offset with respect to the reference year (defined in software).
The year is considered a leap year if YEAR[1:0] is zero.

**Bits 25:22 – MONTH[3:0]** Month
1 – January
2 – February
...
12 – December

**Bits 21:17 – DAY[4:0]** Day
Day starts at 1 and ends at 28, 29, 30, or 31, depending on the month and year.

**Bits 16:12 – HOUR[4:0]** Hour
When CTRLA.CLKREP=0, the Hour bit group is in 24-hour format, with values 0-23. When CTRLA.CLKREP=1, HOUR[3:0] has values 1-12, and HOUR[4] represents AM (0) or PM (1).

**Bits 11:6 – MINUTE[5:0]** Minute
0 – 59

**Bits 5:0 – SECOND[5:0]** Second
0 – 59


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.11. Alarm n Value in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** ALARM
**Offset:** 0x20 + n*0x08 [n=0..1]
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

The 32-bit value of ALARMn is continuously compared with the 32-bit CLOCK value, based on the masking set by MASKn.SEL. When a match occurs, the Alarm n interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.ALARMn) is set on the next counter cycle, and the counter is cleared if CTRLA.MATCHCLR is '1'.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | YEAR[5:0] | | | | | | MONTH[3:2] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:26 – YEAR[5:0]** Year
The alarm year. Years are only matched if MASKn.SEL is 6

**Bits 25:22 – MONTH[3:0]** Month
The alarm month. Months are matched only if MASKn.SEL is greater than 4.

**Bits 21:17 – DAY[4:0]** Day
The alarm day. Days are matched only if MASKn.SEL is greater than 3.

**Bits 16:12 – HOUR[4:0]** Hour
The alarm hour. Hours are matched only if MASKn.SEL is greater than 2.

**Bits 11:6 – MINUTE[5:0]** Minute
The alarm minute. Minutes are matched only if MASKn.SEL is greater than 1.

**Bits 5:0 – SECOND[5:0]** Second
The alarm second. Seconds are matched only if MASKn.SEL is greater than 0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.12. Alarm n Mask in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** MASK
**Offset:** 0x24 + n*0x08 [n=0..1]
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SEL[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 2:0 – SEL[2:0]** Alarm Mask Selection
These bits define which bit groups of Alarm n are valid.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Alarm Disabled |
| 0x1 | SS | Match seconds only |
| 0x2 | MMSS | Match seconds and minutes only |
| 0x3 | HHMMSS | Match seconds, minutes, and hours only |
| 0x4 | DDHHMMSS | Match seconds, minutes, hours, and days only |
| 0x5 | MMDDHHMMSS | Match seconds, minutes, hours, days, and months only |
| 0x6 | YYMMDDHHMMSS | Match seconds, minutes, hours, days, months, and years |
| 0x7 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.13. General Purpose n

**Name:** GPn
**Offset:** 0x40 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | GP[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | GP[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | GP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – GP[31:0]** General Purpose
These bits are for user-defined general purpose use, see General Purpose Registers.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.14. Tamper Control

**Name:** TAMPCTRL
**Offset:** 0x60
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | IN4ACT[1:0] | |
| Access | | | | | | | | |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[1:0] | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 24, 25, 26, 27, 28 – DEBNC** Debounce Enable of Tamper Input INn
**Note:** Debounce feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

| Value | Description |
|---|---|
| 0 | Debouncing is disabled for Tamper input INn |
| 1 | Debouncing is enabled for Tamper input INn |

**Bits 16, 17, 18, 19, 20 – TAMLVL** Tamper Level Select of Tamper Input INn
**Note:** Tamper Level feature does not apply to the Active Layer Protection mode (TAMPCTRL.INACT = ACTL).

| Value | Description |
|---|---|
| 0 | A falling edge condition will be detected on Tamper input INn. |
| 1 | A rising edge condition will be detected on Tamper input INn. |

**Bits 0:1, 2:3, 4:5, 6:7, 8:9 – INACT** Tamper Channel n Action
These bits determine the action taken by Tamper Channel n.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Off (Disabled) |
| 0x1 | WAKE | Wake and set Tamper flag |
| 0x2 | CAPTURE | Capture timestamp and set Tamper flag |
| 0x3 | ACTL | Compare RTC signal routed between INn and OUT pins . When a mismatch occurs, capture timestamp and set Tamper flag |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.15. Timestamp Value

**Name:** TIMESTAMP
**Offset:** 0x64
**Reset:** 0
**Property:** R

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | YEAR[5:0] | | | | | | MONTH[3:2] | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:26 – YEAR[5:0]** Year
The year value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 25:22 – MONTH[3:0]** Month
The month value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 21:17 – DAY[4:0]** Day
The day value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 16:12 – HOUR[4:0]** Hour
The hour value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 11:6 – MINUTE[5:0]** Minute
The minute value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 5:0 – SECOND[5:0]** Second
The second value is captured by the TIMESTAMP when a tamper condition occurs.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.16. Tamper ID

**Name:** TAMPID
**Offset:** 0x68
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TAMPEVT | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | TAMPID4 | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – TAMPEVT** Tamper Event Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

| Value | Description |
|---|---|
| 0 | A tamper input event has not been detected |
| 1 | A tamper input event has been detected |

**Bits 0, 1, 2, 3, 4 – TAMPID** Tamper on Channel n Detected
Writing a '0' to this bit has no effect. Writing a '1' to this bit clears the tamper detection bit.

| Value | Description |
|---|---|
| 0 | A tamper condition has not been detected on Channel n |
| 1 | A tamper condition has been detected on Channel n |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 21.12.17. Backup n

**Name:** BKUP
**Offset:** 0x80 + n*0x04 [n=0..7]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BKUP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – BKUP[31:0]** Backup
These bits are user-defined for general purpose use in the Backup domain.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 22. DMAC – Direct Memory Access Controller

## 22.1. Overview

The Direct Memory Access Controller (DMAC) contains both a Direct Memory Access engine and a Cyclic Redundancy Check (CRC) engine. The DMAC can transfer data between memories and peripherals, and thus off-load these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. With access to all peripherals, the DMAC can handle automatic transfer of data between communication modules.

The DMA part of the DMAC has several DMA channels which can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter (see the Block Diagram). The arbiter will select one DMA channel at a time to act as the active channel. When an active channel has been granted, the fetch engine of the DMAC will fetch a transfer descriptor from the SRAM and store it in the internal memory of the active channel, which will then execute the data transmission.

An ongoing data transfer of an active channel can be interrupted by a higher prioritized DMA channel. The DMAC will write back the updated transfer descriptor from the internal memory of the active channel to SRAM, and grant the higher prioritized channel to start transfer as the new active channel. Once a DMA channel is done with its transfer, interrupts and events can be generated optionally.

The DMAC has four bus interfaces:

- The *data transfer bus* is used for performing the actual DMA transfer
- The *AHB/APB Bridge bus* is used when writing and reading the I/O registers of the DMAC
- The *descriptor fetch bus* is used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued
- The *write-back bus* is used to write the transfer descriptor back to SRAM

All buses are AHB Host interfaces except for the AHB/APB Bridge bus, which is an APB Client interface.

Burst transfer options, buffered active channel to pre-fetch descriptors and advance quality of service features ensure low-latency transfers for high-speed peripherals or high-speed operations.

The CRC engine can be used by software to detect an accidental error in the transferred data and to take corrective action, such as requesting the data to be sent again or simply not using the incorrect data.

## 22.2. Features

- Data transfer from:
  - Peripheral to peripheral
  - Peripheral to memory
  - Memory to peripheral
  - Memory to memory
- Transfer trigger sources
  - Software
  - Events from Event System
  - Dedicated requests from peripherals
- SRAM based transfer descriptors
  - Single transfer using one descriptor

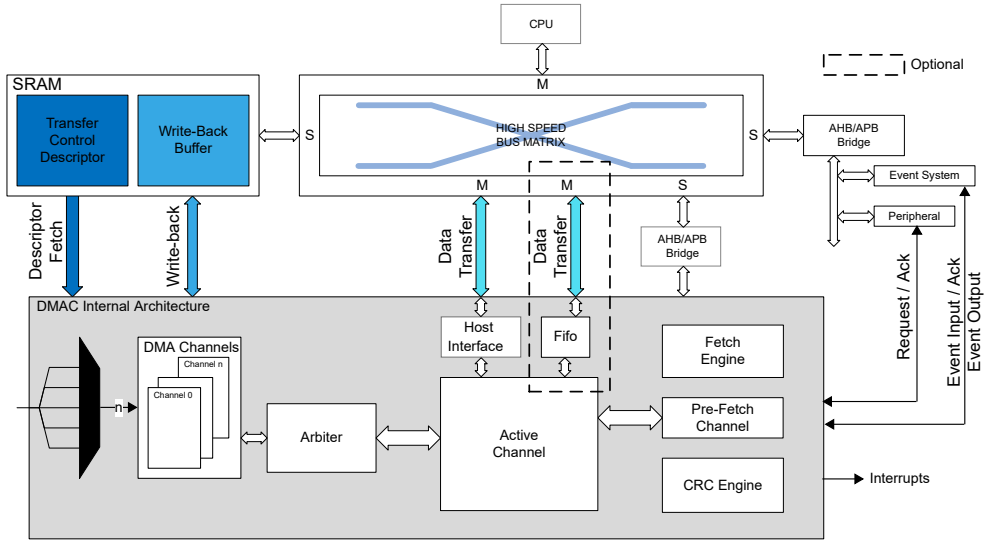

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 32channels
  - Enable 32 independent transfers
  - Automatic descriptor fetch for each channel
  - Suspend/resume operation support for each channel
- Flexible arbitration scheme
  - 4 configurable priority levels for each channel
  - Fixed or round-robin priority scheme within each priority level
- From 1 to 256KB data transfer in a single block transfer
- Multiple addressing modes
  - Static
  - Configurable increment scheme
- Optional interrupt generation
  - On block transfer complete
  - On error detection
  - On channel suspend
- 8 event inputs
  - One event input for each of the 8 least significant DMA channels
  - Can be selected to trigger normal transfers, periodic transfers or conditional transfers
  - Can be selected to suspend or resume channel operation
- 4 event outputs
  - One output event for each of the 4 least significant DMA channels
  - Selectable generation on AHB, block, or transaction transfer complete
- Error management supported by write-back function
  - Dedicated Write-Back memory section for each channel to store ongoing descriptor transfer
- CRC polynomial software selectable to
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE® 802.3)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 22.3. Block Diagram

**Figure 22-1.** DMAC Block Diagram




## 22.4. Signal Description

Not applicable.

## 22.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 22.5.1. I/O Lines

Not applicable.

### 22.5.2. Power Management

The DMAC will continue to operate in any Sleep mode where the selected source clock is running. The DMAC's interrupts can be used to wake-up the device from Sleep modes. Events connected to the event system can trigger other operations in the system without exiting Sleep modes. On hardware or software Reset, all registers are set to their Reset value.

**Related Links**

PM – Power Manager

### 22.5.3. Clocks

An AHB clock (CLK_DMAC_AHB) is required to clock the DMAC. This clock can be configured in the Main Clock peripheral (MCLK) before using the DMAC, and the default state of CLK_DMAC_AHB can be found in the MCLK.AHBMASK register.

**Related Links**

Peripheral Clock Masking


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.5.4. DMA

Not applicable.

### 22.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 22.5.6. Events

The events are connected to the event system.

### 22.5.7. Debug Operation

When the CPU is halted in Debug mode the DMAC will halt normal operation. The DMAC can be forced to continue operation during debugging. Refer to DBGCTRL for details.

### 22.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Pending register (INTPEND)
- Channel Interrupt Flag Status and Clear register (CHINTFLAG)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

### 22.5.9. Analog Connections

Not applicable.

## 22.6. Functional Description

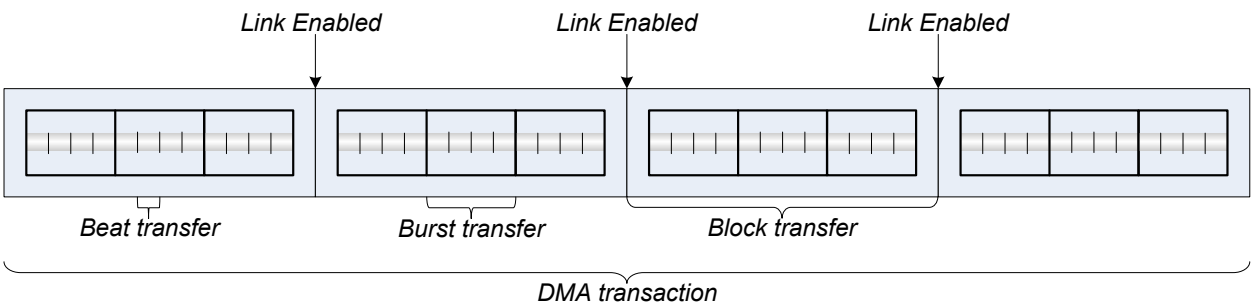
### 22.6.1. Principle of Operation

The DMAC consists of a DMA module and a CRC module.

#### 22.6.1.1. DMA

The DMAC can transfer data between memories and peripherals without interaction from the CPU. The data transferred by the DMAC are called transactions, and these transactions can be split into smaller data transfers. The following figure shows the relationship between the different transfer sizes:

**Figure 22-2.** DMA Transfer Sizes




- Beat Transfer: The size of one data transfer bus access, and the size is selected by writing the Beat Size bit group in the Block Transfer Control register (BTCTRL.BEATSIZE)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Block Transfer: The amount of data one transfer descriptor can transfer, and the amount can range from 1 to 64k beats. A block transfer can be interrupted.
- Burst Transfer: Back-to-back beat transfers without CPU interference.
- Transaction: The DMAC can link several transfer descriptors by having the first descriptor pointing to the second and so forth, as shown in the figure above. A DMA transaction is the complete transfer of all blocks within a linked list.

A transfer descriptor describes how a block transfer should be carried out by the DMAC, and it must remain in SRAM. For further details on the transfer descriptor refer to Transfer Descriptors.

The figure above shows several block transfers linked together, which are called linked descriptors. For further information about linked descriptors, refer to Linked Descriptors.

A DMA transfer is initiated by an incoming transfer trigger on one of the DMA channels. This trigger can be configured to be either a software trigger, an event trigger, or one of the dedicated peripheral triggers. The transfer trigger will result in a DMA transfer request from the specific channel to the arbiter. If there are several DMA channels with pending transfer requests, the arbiter chooses which channel is granted access to become the active channel. The DMA channel granted access as the active channel will carry out the transaction as configured in the transfer descriptor. A current transaction can be interrupted by a higher prioritized channel, but will resume the block transfer when the according DMA channel is granted access as the active channel again.

For each beat transfer, an optional output event can be generated. For each block transfer, optional interrupts and an optional output event can be generated. When a transaction is completed, dependent on the configuration, the DMA channel will either be suspended or disabled.

#### 22.6.1.2. CRC

The internal CRC engine supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). It can be used on a selectable DMA channel, or on the I/O interface. Refer to CRC Operation for details.

### 22.6.2. Basic Operation

#### 22.6.2.1. Initialization

**DMAC Initialization**

Before DMAC is enabled, it must be configured as defined below:

- The SRAM address of where the descriptor memory section is located must be written to the Description Base Address (BASEADDR) register.
- The SRAM address of where the write-back section should be located must be written to the Write-Back Memory Base Address (WRBADDR) register.
- Priority level x of the arbiter can be enabled by setting the Priority Level x Enable bit in the Control register (CTRL.LVLENx=1)

**DMA Channel Initialization**

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as defined below:

- DMA Channel Configuration:
  - The channel number of the DMA channel to configure must be written to the Channel Control A (CHCTRLA) register.
  - Trigger action must be selected by writing the Trigger Action bit field in the Channel Control A (CHCTRLA.TRIGACT) register.
  - Trigger source must be selected by writing the Trigger Source bit field in the Channel Control A (CHCTRLA.TRIGSRC) register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Transfer Descriptor
  - The size of each access of the data transfer bus must be selected by writing the Beat Size bit group in the Block Transfer Control (BTCTRL.BEATSIZE) register.
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control (BTCTRL.VALID) register.
  - Number of beats in the block transfer must be selected by writing the Block Transfer Count (BTCNT) register.
  - Source address for the block transfer must be selected by writing the Block Transfer Source Address (SRCADDR) register.
  - Destination address for the block transfer must be selected by writing the Block Transfer Destination Address (DSTADDR) register.

**CRC Calculation**

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as described below:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control (CRCCTRL.CRCSRC) register.
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control (CRCCTRL.CRCPOLY) register.
- If I/O is selected as input source, the beat size must be selected by writing the CRC Beat Size bit group in the CRC Control (CRCCTRL.CRCBEATSIZE) register.

**Register Properties**

The following DMAC registers are enable-protected, that is, they can only be written when the DMAC is disabled (CTRL.DMAENABLE=0):

- The Descriptor Base Memory Address (BASEADDR) register
- The Write-Back Memory Base Address (WRBADDR) register

The following DMAC bit is enable-protected, that is, it can only be written when the DMAC and CRC are disabled (CTRL.DMAENABLE=0 and CRCCTRL.CRCSRC=0):

- The Software Reset bit in the Control (CTRL.SWRST) register

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- The Channel Software Reset bit in the Channel Control A (CHCTRLA.SWRST) register

The following CRC registers are enable-protected, that is, they can only be written when the CRC is disabled (CRCCTRL.CRCSRC=0):

- The CRC Control (CRCCTRL) register
- CRC Checksum (CRCCHKSUM) register

Enable-protection is denoted by the ‘Enable-Protected’ property in the register description.

### 22.6.2.2. Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control (CTRL.DMAENABLE) register to '1'. The DMAC is disabled by writing a '0' to the CTRL.DMAENABLE register.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after the corresponding channel ID to the channel is configured. A DMA channel is disabled by writing a '0' to CHCTRLAn.ENABLE.

The CRC is enabled by writing a value to the CRC Source bits in the Control register (CRCCTRL.CRCSRC). The CRC is disabled by writing a '0' to CRCCTRL.CRCSRC.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The DMAC is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST) while the DMAC and CRC are disabled. All registers in the DMAC except DBGCTRL will be reset to their initial state.

A DMA channel is reset by writing a '1' to the Software Reset bit in the Channel Control A register (CHCTRLAn.SWRST), after the corresponding channel is configured. The channel registers will be reset to their initial state. The corresponding DMA channel must be disabled in order for the Reset to take effect.

#### 22.6.2.3. Transfer Descriptors

The transfer descriptors, together with the channel configurations, decide how a block transfer should be executed. Before a DMA channel is enabled (CHCTRLA.ENABLE is written to one) and receives a transfer trigger, its first transfer descriptor must be initialized and valid (BTCTRL.VALID). The first transfer descriptor describes the first block transfer of a transaction.

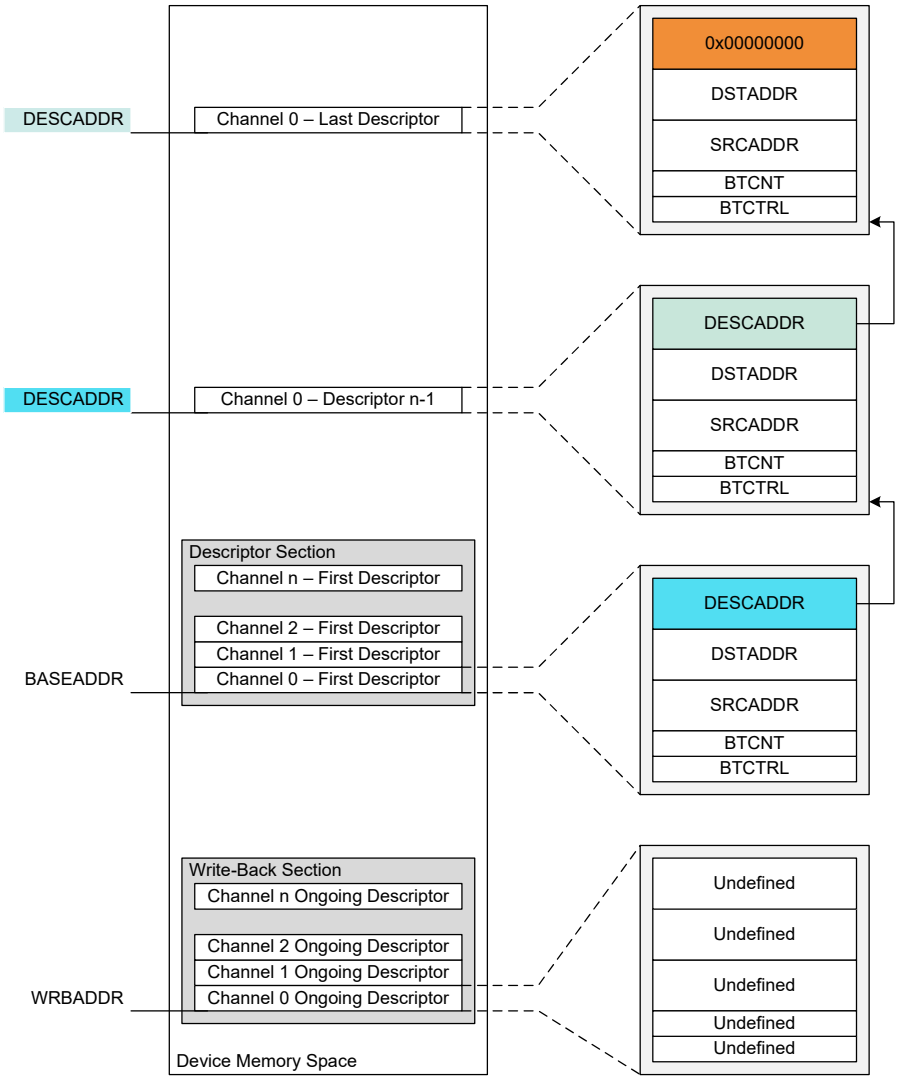
All transfer descriptors must reside in SRAM. The addresses stored in the Descriptor Memory Section Base Address (BASEADDR) and Write-Back Memory Section Base Address (WRBADDR) registers tell the DMAC where to find the descriptor memory section and the write-back memory section.

The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel '0' (see figure below). All first transfer descriptors must be stored in a contiguous memory section, where the transfer descriptors must be ordered according to their channel number. For further details on linked descriptors, refer to Linked Descriptors.

The write-back memory section is where the DMAC stores the transfer descriptors for the ongoing block transfers. WRBADDR points to the ongoing transfer descriptor of channel '0'. All ongoing transfer descriptors are stored in a contiguous memory section where the transfer descriptors are ordered according to their channel number. The figure below shows an example of linked descriptors on DMA channel '0'. For additional information on linked descriptors, refer to the Linked Descriptors.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-3.** Memory Sections




The size of the descriptor and write-back memory sections are dependent on the number of the most significant enabled DMA channel *m*, as shown below:

$Size = 128\text{bits} \cdot (m + 1)$

For memory optimization, it is recommended to use the less significant DMA channels, if not all channels are required.

The descriptor and write-back memory sections can either be two separate memory sections, or they can share a memory section (BASEADDR=WRBADDR). The benefit of having them in two separate sections, is that the same transaction for a channel can be repeated without having to modify the first transfer descriptor. In addition, the latency from fetching the first descriptor of a transaction to the first burst transfer is executed, is reduced.

### 22.6.2.4. Arbitration

If a DMA channel is enabled and not suspended when it receives a transfer trigger, it will send a transfer request to the arbiter. When the arbiter receives the transfer request it will include the DMA channel in the queue of channels having pending transfers, and the corresponding Pending Channel


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

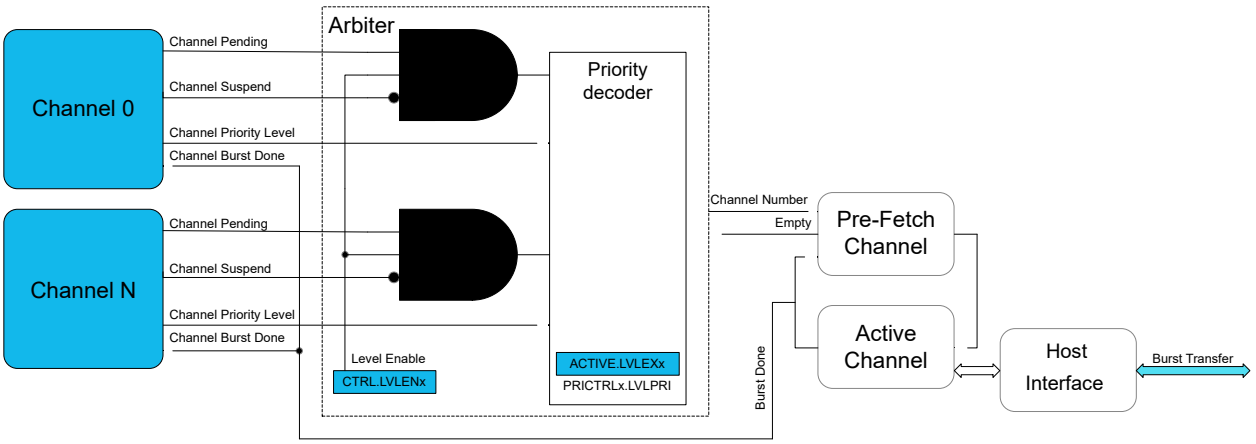
x bit in the Pending Channels registers (PENDCH.PENDCHx) will be set. Depending on the arbitration scheme, the arbiter will choose which DMA channel will be the next active channel. The next transfer descriptor will be fetched from SRAM memory and stored internally in the Pre-Fetch Channel. The active channel is the DMA channel being granted access to perform its next burst transfer. When the Active Channel has completed a burst transfer, the descriptor stored in the Pre-Fetch Channel is transferred to the Active Channel and a new burst will take place.

When the descriptor stored in the Pre-Fetch Channel is transferred to the Active Channel, the corresponding PENDCH.PENDCHx will be cleared. In the same way, depending on trigger action settings and if the upcoming burst transfer is the first for the transfer request or not, the corresponding Busy Channel x bit in the Busy Channels register (BUSYCH.BUSYCHx), will either be set or remain '1'. When the channel has performed its granted burst transfer(s) it will be either fed into the queue of channels with pending transfers, set to be waiting for a new transfer trigger, suspended, or disabled. This depends on the channel and block transfer configuration. If the DMA channel is set to wait for a new transfer trigger, suspended or disabled, the corresponding BUSYCH.BUSYCHx will be cleared.

If a DMA channel is suspended while it has a pending transfer, it will be removed from the queue of pending channels, but the corresponding PENDCH.PENDCHx will remain set. The status will also be indicated in CHINTFLAGn.SUSP. When the same DMA channel is resumed, it will be added to the queue of pending channels again.

If a DMA channel gets disabled (CHCTRLA.ENABLE=0) while it has a pending transfer, it will be removed from the queue of pending channels, and the corresponding PENDCH.PENDCHx will be cleared.

**Figure 22-4.** Arbiter Overview




### Priority Levels

When a channel level is pending or the channel is transferring data, the corresponding Level Executing bit is set in the Active Channel and Levels register (ACTIVE.LVLEXx).

Each DMA channel supports up to4-level priority scheme. The number of supported priority levels will differ from one device family to another.

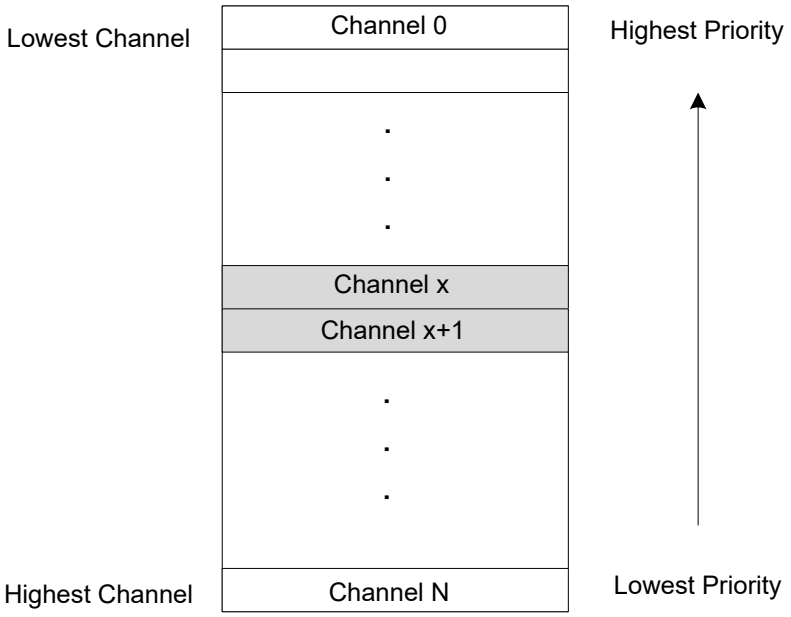
The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Priority Level register (CHPRILVL.PRILVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority level number. A priority level is enabled by writing the Priority Level x Enable bit in the Control register (CTRL.LVLENx) to '1', for the corresponding level.

Within each priority level, the DMAC's arbiter can be configured to prioritize statically or dynamically. For the arbiter to perform static arbitration within a priority level, the Level X Round-Robin


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Scheduling Enable bit in the Priority Control x register (PRICTRL0.RRLVLENx) has to be written to '0'. When static arbitration is enabled (PRICTRL0.RRLVLENx is '0'), the arbiter will prioritize a low channel number over a high channel number as shown in Static Priority Scheduling. When using the static scheme, there is a risk of high channel numbers never being granted access as the active channel. This can be avoided using a dynamic arbitration scheme.

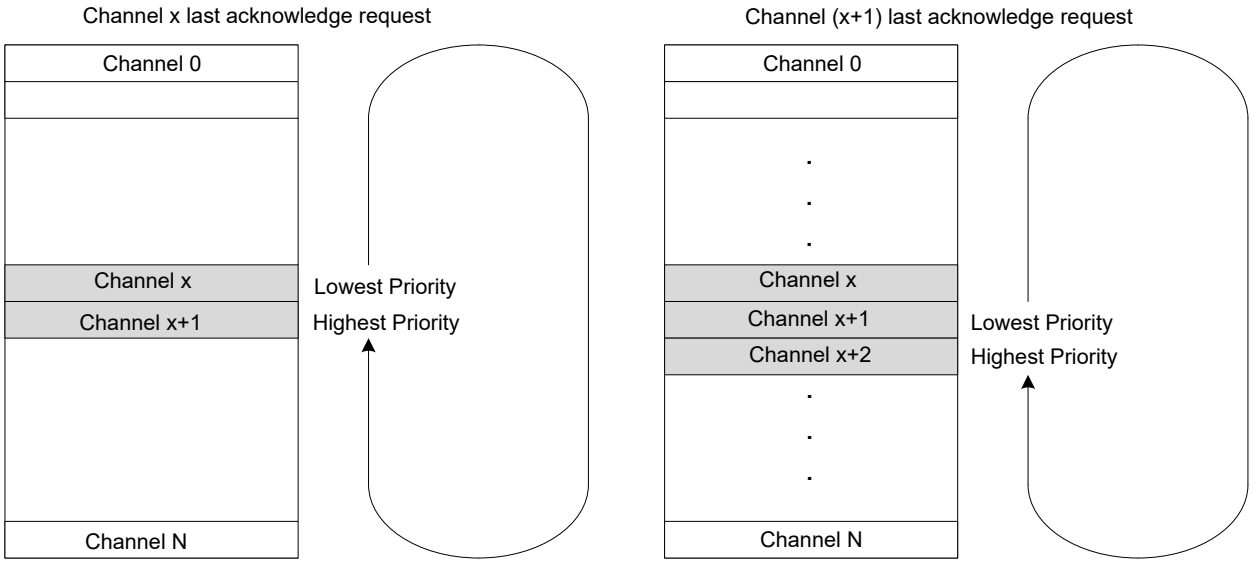
**Figure 22-5.** Static Priority Scheduling




The dynamic arbitration scheme in the DMAC is round-robin. Round-robin arbitration is enabled by writing PRICTRL0.RRLVLENx to '1', for a given priority level x. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel within the same priority level, as shown in *Dynamic (Round-Robin) Priority Scheduling*. The channel number of the last channel being granted access as the active channel is stored in the Level x Channel Priority Number bit group in the Priority Control 0 register (PRICTRL0.LVLPRIx) for the corresponding priority level.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-6.** Dynamic (Round-Robin) Priority Scheduling




### 22.6.2.5. Data Transmission

Before the DMAC can perform a data transmission, a DMA channel has to be configured and enabled, its corresponding transfer descriptor has to be initialized, and the arbiter has to grant the DMA channel access as the active channel.

Once the arbiter has granted a DMA channel access as the active channel (refer to DMA Block Diagram section) the transfer descriptor for the DMA channel will be fetched from SRAM using the fetch bus, and stored in the internal memory for the active channel. For a new block transfer, the transfer descriptor will be fetched from the descriptor memory section (BASEADDR); For an ongoing block transfer, the descriptor will be fetched from the write-back memory section (WRBADDR). By using the data transfer bus, the DMAC will read the data from the current source address and write it to the current destination address. For further details on how the current source and destination addresses are calculated, refer to the section on Addressing.

The arbitration procedure is performed after each burst transfer. If the current DMA channel is granted access again, the block transfer counter (BTCNT) of the internal transfer descriptor will be decremented by the number of beats in a burst transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new burst transfer. If a different DMA channel than the current active channel is granted access, the block transfer counter value will be written to the write-back section before the transfer descriptor of the newly granted DMA channel is fetched into the internal memory of the active channel.

When a block transfer has come to its end (BTCNT is zero), the Valid bit in the Block Transfer Control register will be cleared (BTCTRL.VALID=0) before the entire transfer descriptor is written to the write-back memory. The optional interrupts, Channel Transfer Complete and Channel Suspend, and the optional output event Block, will be generated if configured and enabled. After the last block transfer in a transaction, the Next Descriptor Address register (DESCADDR) will hold the value 0x00000000, and the DMA channel will either be suspended or disabled, depending on the configuration in the Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT). If the transaction has further block transfers pending, DESCADDR will hold the SRAM address to the next transfer descriptor to be fetched. The DMAC will fetch the next descriptor into the internal memory of the active channel and write its content to the write-back section for the channel, before the arbiter gets to choose the next active channel.

**Related Links**

Block Diagram


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

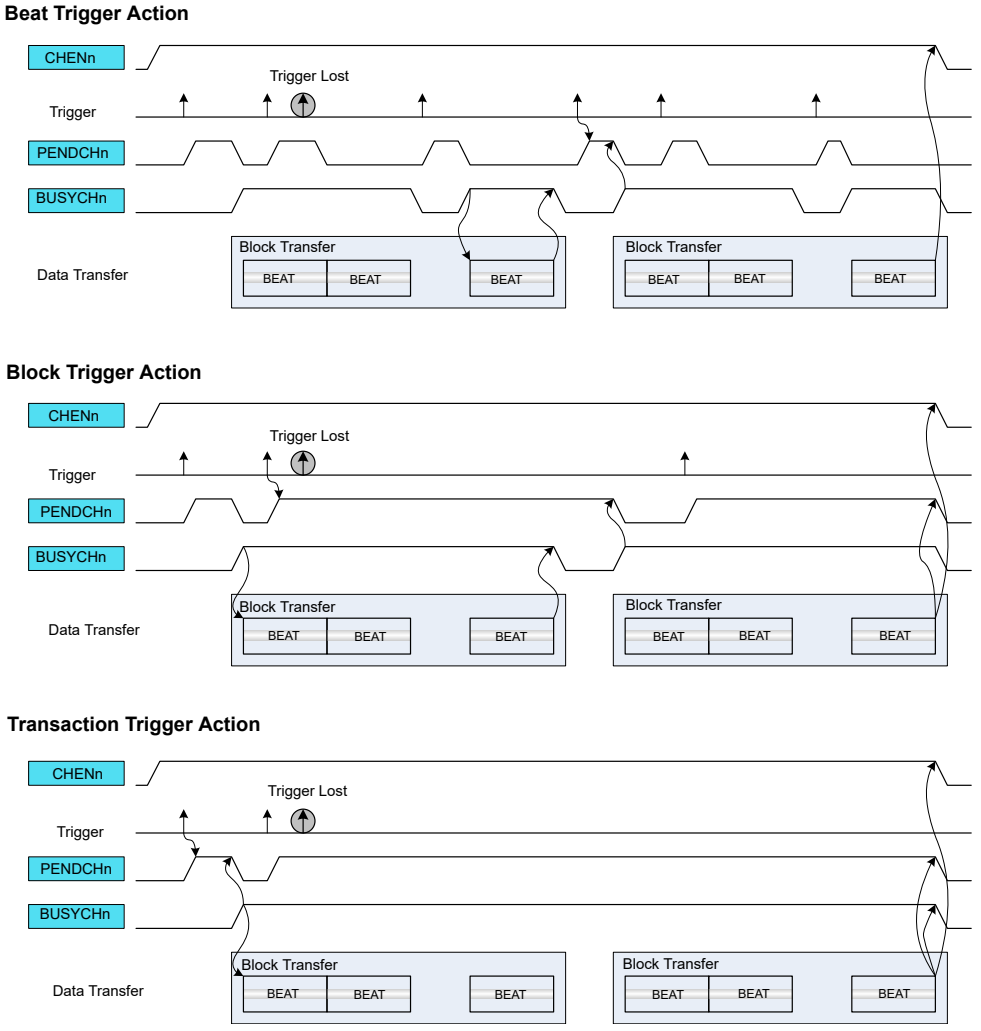
### 22.6.2.6. Transfer Triggers and Actions

A DMA transfer through a DMA channel can be started only when a DMA transfer request is detected, and the DMA channel has been granted access to the DMA. A transfer request can be triggered from software, from a peripheral, or from an event. There are dedicated Trigger Source selections for each DMA Channel n Control A (CHCTRLAn.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel n Control A register (CHCTRLAn.TRIGACT). By default, a trigger generates a request for a block transfer operation. If a single descriptor is defined for a channel, the channel is automatically disabled when a block transfer has been completed. If a list of linked descriptors is defined for a channel, the channel is automatically disabled when the last descriptor in the list is executed. As long as the list still has descriptors to execute, the channel will be waiting for the next block transfer trigger. When enabled again, the channel will wait for the next block transfer trigger. The trigger actions can also be configured to generate a request for a burst transfer (CHCTRLAn.TRIGACT=0x2) or transaction transfer (CHCTRLAn.TRIGACT=0x3) instead of a block transfer (CHCTRLAn.TRIGACT=0x0).

The following figure shows an example where triggers are used with two linked block descriptors.

**Figure 22-7.** Trigger Action and Transfers




If the trigger source generates a transfer request for a channel during an ongoing transfer, the new transfer request will be kept pending (CHSTATUSn.PEND=1), and the new transfer can start after the ongoing one is done. Only one pending transfer can be kept per channel. If the trigger source


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

generates more transfer requests while one is already pending, the additional ones will be lost. All channels pending status flags are also available in the Pending Channels register (PENDCH).

When the transfer starts, the corresponding Channel Busy status flag is set in Channel n Status register (CHSTATUSn.BUSY). When the trigger action is complete, the Channel Busy status flag is cleared. All channel busy status flags are also available in the Busy Channels register (BUSYCH) in DMAC.

#### 22.6.2.7. Addressing

Each block transfer must have a source address and a destination address defined. The source address is set by writing the Transfer Source Address (SRCADDR) register, the destination address is set by writing the Transfer Destination Address (DSTADDR) register.

The addressing of this DMAC module can be static or incremental, for either source or destination of a block transfer, or both.

Incrementation for the source address of a block transfer is enabled by writing the Source Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.SRCINC = 1). The step size of the incrementation is configurable and can be chosen by writing the Step Selection bit in the Block Transfer Control register (BTCTRL.STEPSEL = 1) and writing the desired step size in the Address Increment Step Size bit group in the Block Transfer Control register (BTCTRL.STEPSIZE). If BTCTRL.STEPSEL = 0, the step size for the source incrementation will be the size of one beat.

When source address incrementation is configured (BTCTRL.SRCINC = 1), SRCADDR is calculated as follows:

If BTCTRL.STEPSEL= 1:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE\right) \cdot 2^{\mathrm{STEPSIZE}}$$

If BTCTRL.STEPSEL= 0:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE\right)$$

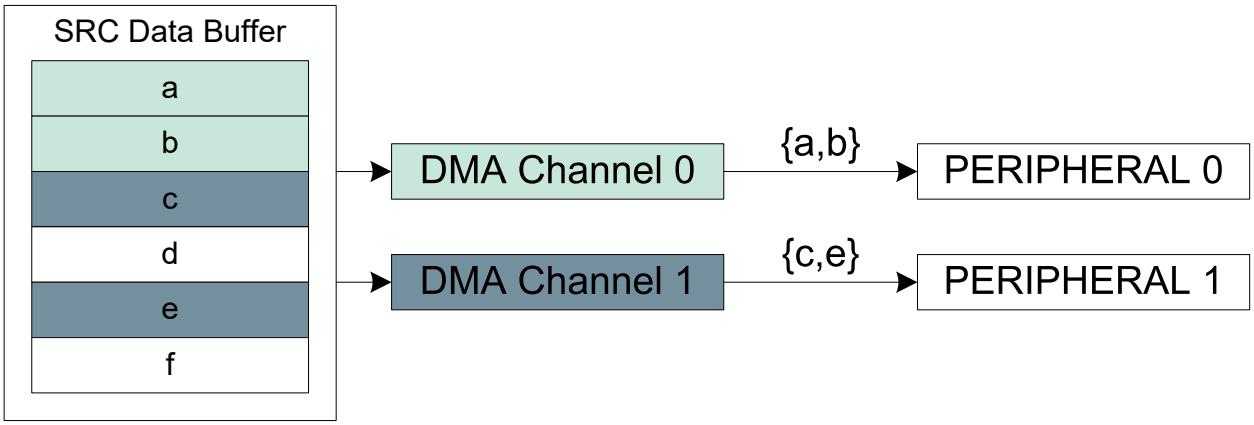
Where,

- $\mathrm{SRCADDR}_{\mathrm{START}}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat (1, 2, or 4 bytes)
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment the source address by one beat after each beat transfer (BTCTRL.SRCINC = 1), and DMA channel 1 is configured to increment the source address by two beats (BTCTRL.SRCINC = 1, BTCTRL.STEPSEL = 1, and BTCTRL.STEPSIZE = 0x1). As the destination address for both channels are peripherals, destination incrementation is disabled (BTCTRL.DSTINC = 0).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-8.** Source Address Increment




Incrementation for the destination address of a block transfer is enabled by setting the Destination Address Incrementation Enable bit in the Block Transfer Control register (BTCTRL.DSTINC = 1). The step size of the incrementation is configurable by clearing BTCTRL.STEPSEL= 0 and writing BTCTRL.STEPSIZE to the desired step size. If BTCTRL.STEPSEL= 1, the step size for the destination incrementation will be the size of one beat.

When the destination address incrementation is configured (BTCTRL.DSTINC = 1), DSTADDR must be set and calculated as follows:

| | |
|---|---|
| $DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE\right) \bullet 2^{STEPSIZE}$ | where BTCTRL.STEPSEL is zero |
| $DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE)$ | where BTCTRL.STEPSEL is one |

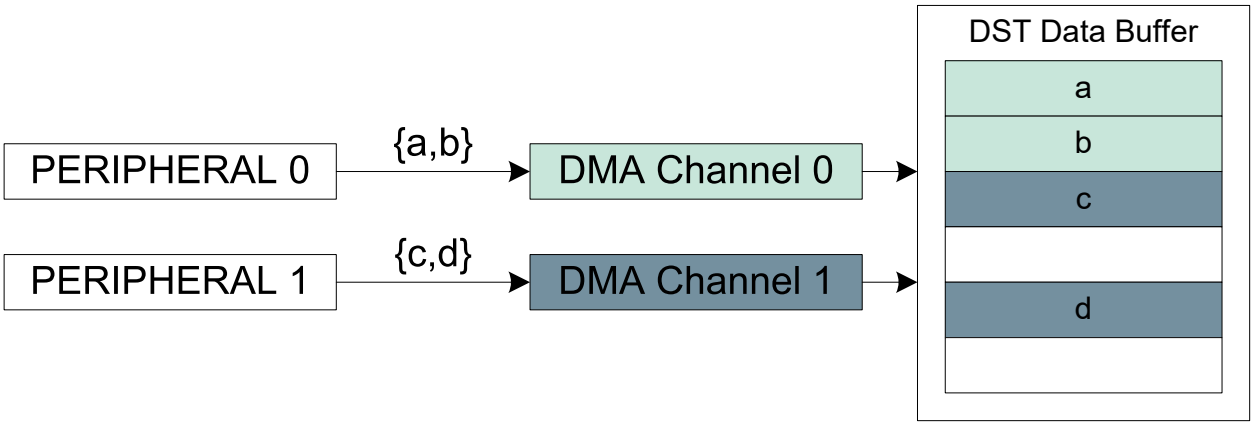
Where,

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat (1, 2, or 4 bytes)
- STEPSIZE is the configured number of beats for each incrementation

The following figure shows an example where DMA channel 0 is configured to increment destination address by one beat (BTCTRL.DSTINC = 1) and DMA channel 1 is configured to increment destination address by two beats (BTCTRL.DSTINC = 1, BTCTRL.STEPSEL= 0, and BTCTRL.STEPSIZE = 0x1). As the source address for both channels are peripherals, source incrementation is disabled (BTCTRL.SRCINC = 0).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-9.** Destination Address Increment




#### 22.6.2.8. Internal FIFO

To improve the bandwidth, the DMAC can support FIFO operation. When single-beat burst configuration is selected (CHCTRALx.BURSTLEN = SINGLE), the channel waits until the FIFO can transmit or accept a single beat transfer before it requests a bus access to write to the destination address. In all other cases, the channel waits until the FIFO threshold is reached before it requests a bus access to write to the destination address. The threshold is configurable and can be set by writing the THRESHOLD bits in the Channel x Control A register.

If the DMAC completes the read operations before the threshold is reached, the write to the destination is automatically enabled. If the FIFO is empty and the read from source is ongoing, the DMA will wait again until the FIFO threshold is reached before it requests a bus access to write the destination.

#### 22.6.2.9. Error Handling

If a bus error is received from an AHB client during a DMA data transfer, the corresponding active channel is disabled and the corresponding Channel Transfer Error Interrupt flag in the Channel Interrupt Status and Clear register (CHINTFLAG.TERR) is set. If enabled, the optional transfer error interrupt is generated. The transfer counter will not be decremented and its current value is written-back in the write-back memory section before the channel is disabled.

When the DMAC fetches an invalid descriptor (BTCTRL.VALID=0) or when the channel is resumed and the DMA fetches the next descriptor with null address (DESCADDR=0x00000000), the corresponding channel operation is suspended, the Channel Suspend Interrupt Flag in the Channel Interrupt Flag Status and Clear register (CHINTFLAG.SUSP) is set, and the Channel Fetch Error bit in the Channel Status register (CHSTATUS.FERR) is set. If enabled, the optional suspend interrupt is generated.

### 22.6.3. Additional Features

#### 22.6.3.1. Linked Descriptors

A transaction can consist of either a single block transfer or of several block transfers. When a transaction consists of several block transfers it is done with the help of linked descriptors.

Figure 22-3 illustrates how linked descriptors work. When the first block transfer is completed on DMA channel 0, the DMAC fetches the next transfer descriptor, which is pointed to by the value stored in the Next Descriptor Address (DESCADDR) register of the first transfer descriptor. Fetching the next transfer descriptor (DESCADDR) is continued until the last transfer descriptor. When the block transfer for the last transfer descriptor is executed and DESCADDR=0x00000000, the transaction is terminated. For further details on how the next descriptor is fetched from SRAM, refer to section Data Transmission.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 22.6.3.1.1. Adding Descriptor to the End of a List

To add a new descriptor at the end of the descriptor list, create the descriptor in SRAM, with DESCADDR=0x00000000 indicating that it is the new last descriptor in the list, and modify the DESCADDR value of the current last descriptor to the address of the newly created descriptor.

#### 22.6.3.1.2. Modifying a Descriptor in a List

In order to add descriptors to a linked list, the following actions must be performed:

1. Enable the Suspend interrupt for the DMA channel.
2. Enable the DMA channel.
3. Reserve memory space in SRAM to configure a new descriptor.
4. Configure the new descriptor:
   - Set the next descriptor address (DESCADDR)
   - Set the destination address (DSTADDR)
   - Set the source address (SRCADDR)
   - Configure the block transfer control (BTCTRL) including
     - Optionally enable the suspend block action
     - Set the descriptor VALID bit
5. Clear the VALID bit for the existing list and for the descriptor which has to be updated.
6. Read DESCADDR from the write-back memory.
   - If the DMA has not already fetched the descriptor that requires changes (i.e., DESCADDR is wrong):
     - Update the DESCADDR location of the descriptor from the list
     - Optionally clear the suspend block action
     - Set the descriptor VALID bit to '1'
     - Optionally enable the Resume Software command
   - If the DMA is executing the same descriptor as the one that requires changes:
     - Set the Channel Suspend Software command and wait for the suspend interrupt
     - Update the next descriptor address (DESCRADDR) in the write-back memory
     - Clear the interrupt sources and set the Resume Software command
     - Update the DESCADDR location of the descriptor from the list
     - Optionally clear the suspend block action
     - Set the descriptor VALID bit to '1'
7. Go to step 4 if needed.

#### 22.6.3.1.3. Adding a Descriptor Between Existing Descriptors

To insert a new descriptor 'C' between two existing descriptors ('A' and 'B'), the descriptor currently executed by the DMA must be identified.

1. If DMA is executing descriptor B, descriptor C cannot be inserted.
2. If DMA has not started to execute descriptor A, follow the steps:
   a. Set the descriptor A VALID bit to '0'.
   b. Set the DESCADDR value of descriptor A to point to descriptor C instead of descriptor B.
   c. Set the DESCADDR value of descriptor C to point to descriptor B.
   d. Set the descriptor A VALID bit to '1'.
3. If DMA is executing descriptor A:

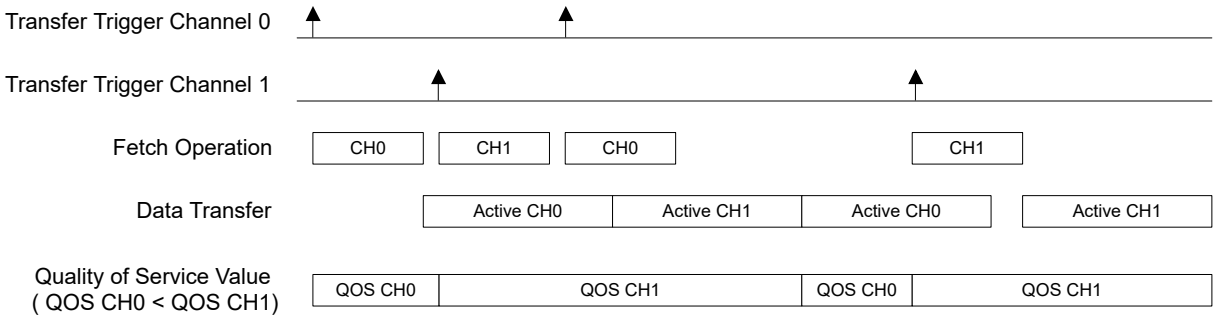

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

a. Apply the software suspend command to the channel and
b. Perform steps 2.1 through 2.4.
c. Apply the software resume command to the channel.

### 22.6.3.2. Transfer Quality of Service

Each priority level group has dedicated quality of service settings. The setting can be written in the corresponding Quality of Service bit group in the Priority Control x register (PRICTRL0.QOSn).

**Figure 22-10.** Quality of Service




When a channel is stored in the Pre-Fetch or Active Channel, the corresponding PRICTRLx.QOS bits value is stored in the respective channel. As shown in Quality of Service, the DMAC will select the highest QOS value between Active and Pre-Fetch channels. This value will apply to all DMAC buses.

### 22.6.3.3. Channel Suspend

The channel operation can be suspended at any time by software by writing a '1' to the Suspend command in the Command bit field of Channel Control B register (CHCTRLB.CMD). After the ongoing burst transfer is completed, the channel operation is suspended and the suspend command is automatically cleared.

When suspended, the Channel Suspend Interrupt flag in the Channel Interrupt Status and Clear register is set (CHINTFLAG.SUSP=1) and the optional suspend interrupt is generated.

By configuring the block action to suspend by writing Block Action bit group in the Block Transfer Control register (BTCTRL.BLOCKACT is 0x2 or 0x3), the DMA channel will be suspended after it has completed a block transfer. The DMA channel will be kept enabled and will be able to receive transfer triggers, but it will be removed from the arbitration scheme.

If an invalid transfer descriptor (BTCTRL.VALID=0) is fetched from SRAM, the DMA channel will be suspended, and the Channel Fetch Error bit in the Channel Status register(CHASTATUS.FERR) will be set.

**Note:** Only enabled DMA channels can be suspended. If a channel is disabled when it is attempted to be suspended, the internal suspend command will be ignored.

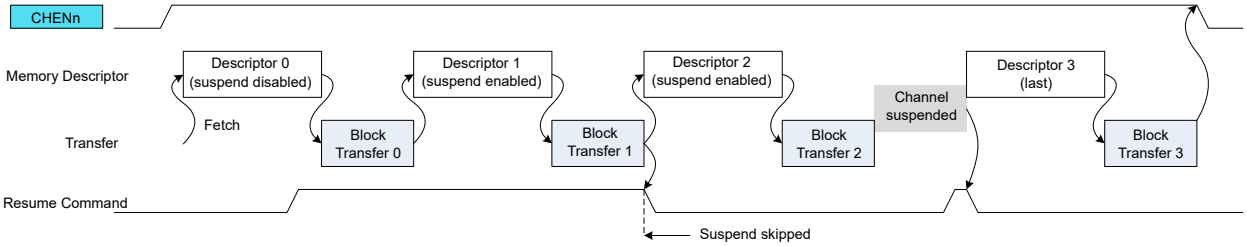
For more details on transfer descriptors, refer to section Transfer Descriptors.

### 22.6.3.4. Channel Resume and Next Suspend Skip

A channel operation can be resumed by software by setting the Resume command in the Command bit field of the Channel Control B register (CHCTRLB.CMD). If the channel is already suspended, the channel operation resumes from where it previously stopped when the Resume command is detected. When the Resume command is issued before the channel is suspended, the next suspend action is skipped and the channel continues the normal operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-11.** Channel Suspend/Resume Operation




### 22.6.3.5. Event Input Actions

The event input actions are available only on the least significant DMA channels. For details on channels with event input support, refer to the Event System documentation.

Before using event input actions, the event controller must be configured first according to the following table, and the Channel Event Input Enable bit in the Channel Event Control register (CHEVCTRL.EVIE) must be written to '1'. Refer also to Events.

**Table 22-1.** Event Input Action

| Action | CHEVCTRL.EVACT | CHCTRLA.TRIGSRC |
|---|---|---|
| None | NOACT | - |
| Normal Transfer | TRIG | DISABLE |
| Conditional Transfer on Strobe | TRIG | Any peripheral |
| Conditional Transfer | CTRIG | |
| Conditional Block Transfer | CBLOCK | |
| Channel Suspend | SUSPEND | |
| Channel Resume | RESUME | |
| Skip Next Block Suspend | SSKIP | |
| Increase priority | INCPRI | |

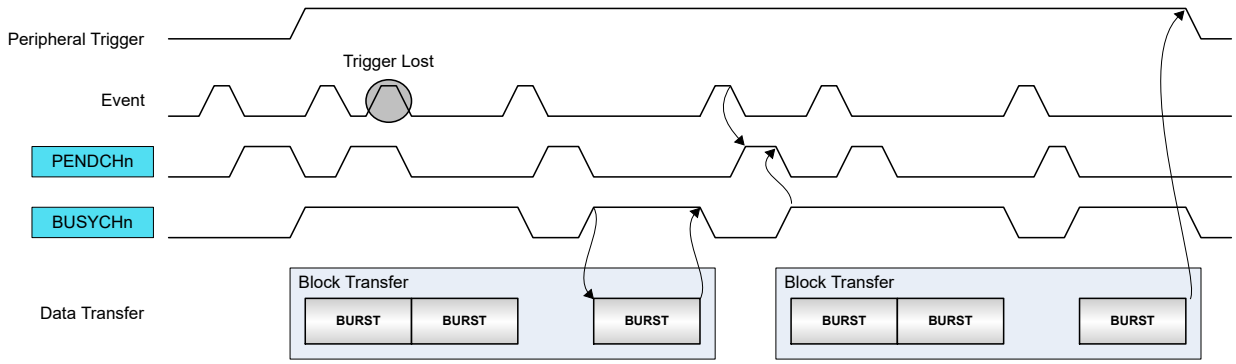
#### Normal Transfer

The event input is used to trigger a beat or burst transfer on peripherals.

The event is acknowledged as soon as the event is received. When received, both the Channel Pending status bit in the Channel Status register (CHSTATUS.PEND) and the corresponding Channel n bit in the Pending Channels register (PENDCH.PENDCHn) are set. If the event is received while the channel is pending, the event trigger is lost.

The figure below shows an example where beat transfers are enabled by internal events.

**Figure 22-12.** Burst Event Trigger Action





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

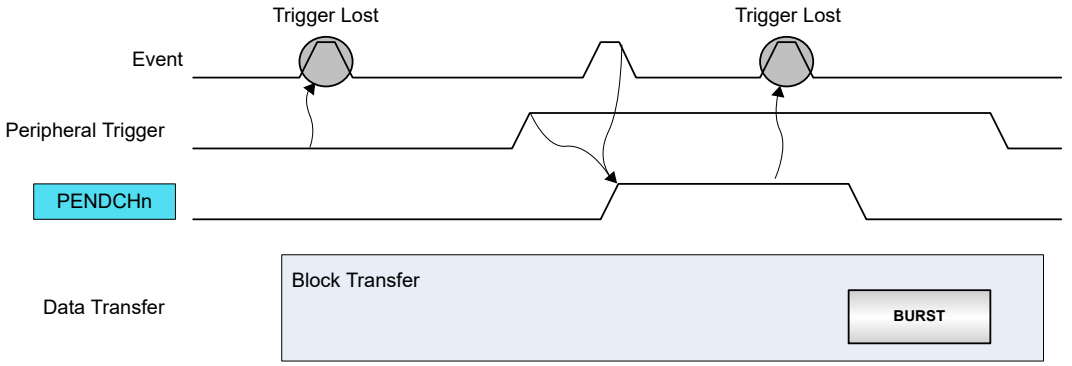
### Conditional Transfer on Strobe

The event input is used to trigger a transfer on peripherals with pending transfer requests. This event action is intended to be used with peripheral triggers, e.g., for timed communication protocols or periodic transfers between peripherals: only when the peripheral trigger coincides with the occurrence of a (possibly cyclic) event the transfer is issued.

The event is acknowledged as soon as the event is received. The peripheral trigger request is stored internally when the previous trigger action is completed (i.e., the channel is not pending) and when an active event is received. If the peripheral trigger is active, the DMA will wait for an event before the peripheral trigger is internally registered. When both event and peripheral transfer trigger are active, both CHSTATUS.PEND and PENDCH.PENDCHn are set. A software trigger will now trigger a transfer.

The figure below shows an example where the peripheral beat transfer is started by a conditional strobe event action.

**Figure 22-13.** Periodic Event with Burst Peripheral Triggers



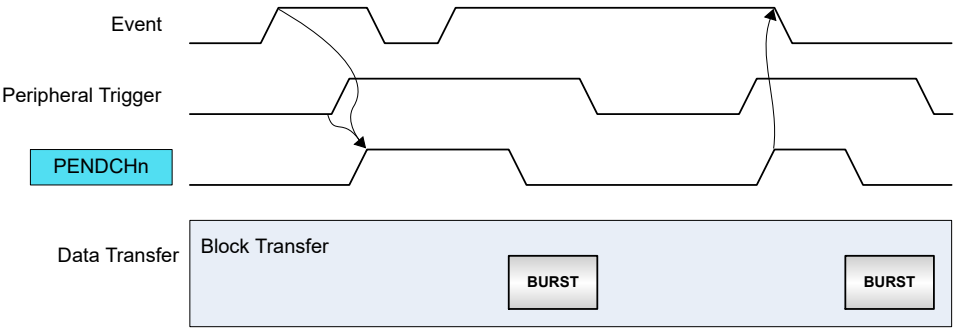

### Conditional Transfer

The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. As example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

Each peripheral trigger is stored internally when the event is received. When the peripheral trigger is stored internally, the Channel Pending status bit is set (CHSTATUS.PEND), the respective Pending Channel n Bit in the Pending Channels register is set (PENDCH.PENDCHn), and the event is acknowledged. A software trigger will now trigger a transfer.

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

**Figure 22-14.** Conditional Event with Burst Peripheral Triggers





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

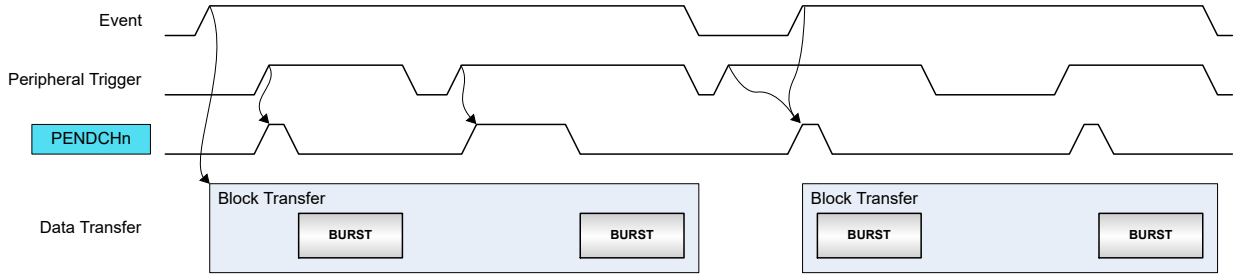
### Conditional Block Transfer

The event input is used to trigger a conditional block transfer on peripherals.

Before starting transfers within a block, an event must be received. When received, the event is acknowledged when the block transfer is completed. A software trigger will trigger a transfer.

The figure below shows an example where conditional event block transfer is started with peripheral beat trigger requests.

**Figure 22-15.** Conditional Block Transfer with Burst Peripheral Triggers




### Channel Suspend

The event input is used to suspend an ongoing channel operation. The event is acknowledged when the current AHB access is completed. For further details on Channel Suspend, refer to Channel Suspend.

### Channel Resume

The event input is used to resume a suspended channel operation. The event is acknowledged as soon as the event is received and the Channel Suspend Interrupt Flag (CHINTFLAG.SUSP) is cleared. For further details refer to Channel Suspend.

### Skip Next Block Suspend

This event can be used to skip the next block suspend action. If the channel is suspended before the event rises, the channel operation is resumed and the event is acknowledged. If the event rises before a suspend block action is detected, the event is kept until the next block suspend detection. When the block transfer is completed, the channel continues the operation (not suspended) and the event is acknowledged.

### Increase priority

This event can be used to increase a channel priority and to request higher quality of service (QOS), when critical transfers must be done. When the event is detected, the channel will have the highest priority and the output Quality of Service value is internally forced to the maximum value. The event is acknowledged when the trigger action execution is completed. When acknowledged, the channel will recover its initial priority level and quality of service settings.

## 22.6.3.6. Event Output Selection

The event output selections are available only for channels supporting event outputs.

The Channel Event Output Enable can be set in the corresponding Channel n Event Control register (CHEVCTRL.EVOE). The Event Output Mode bits in Channel n Event Control register (CHEVCTRL.EVOMODE) selects the event type the channel should generate.

The transfer events (CHEVCTRL.EVOMODE = DEFAULT) are strobe events and their duration is one CLK_DMAC_AHB clock period. The transfer event type selection is available in each Descriptor Block Control location (BTCTRL.EVOSEL). Block or burst event output generation is supported.

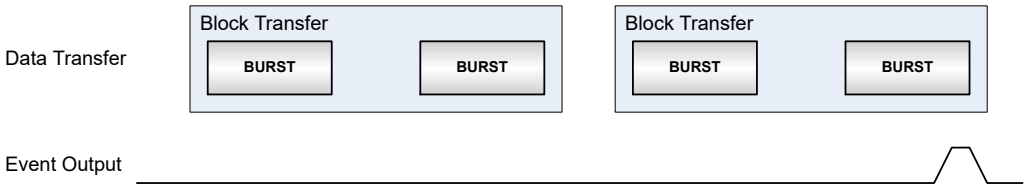
The trigger action event (CHEVCTRL.EVOMODE = TRIGACT) is a level, active while the trigger action execution is not completed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### Block event output

When the block event output is selected, an event strobe is generated when the block transfer is completed. The pulse width of a block event output from a channel is one AHB clock cycle. It is also possible to use this event type to generate an event when the transaction is complete. For this type of application, the block event selection must be set in the last transfer descriptor only, as shown below.

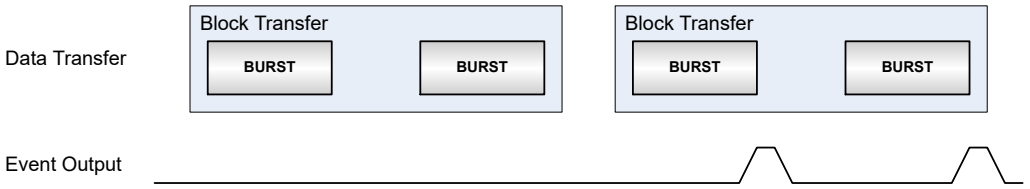
**Figure 22-16.** Block Event Output Generation




### Burst event output

When the burst event output is selected, an event strobe is generated when each burst transfer within the corresponding block is completed. The pulse width of a burst event output from a channel is one AHB clock cycle. The figure below shows an example where the burst event output is set in the second descriptor of a linked list.

**Figure 22-17.** Burst Event Output Generation

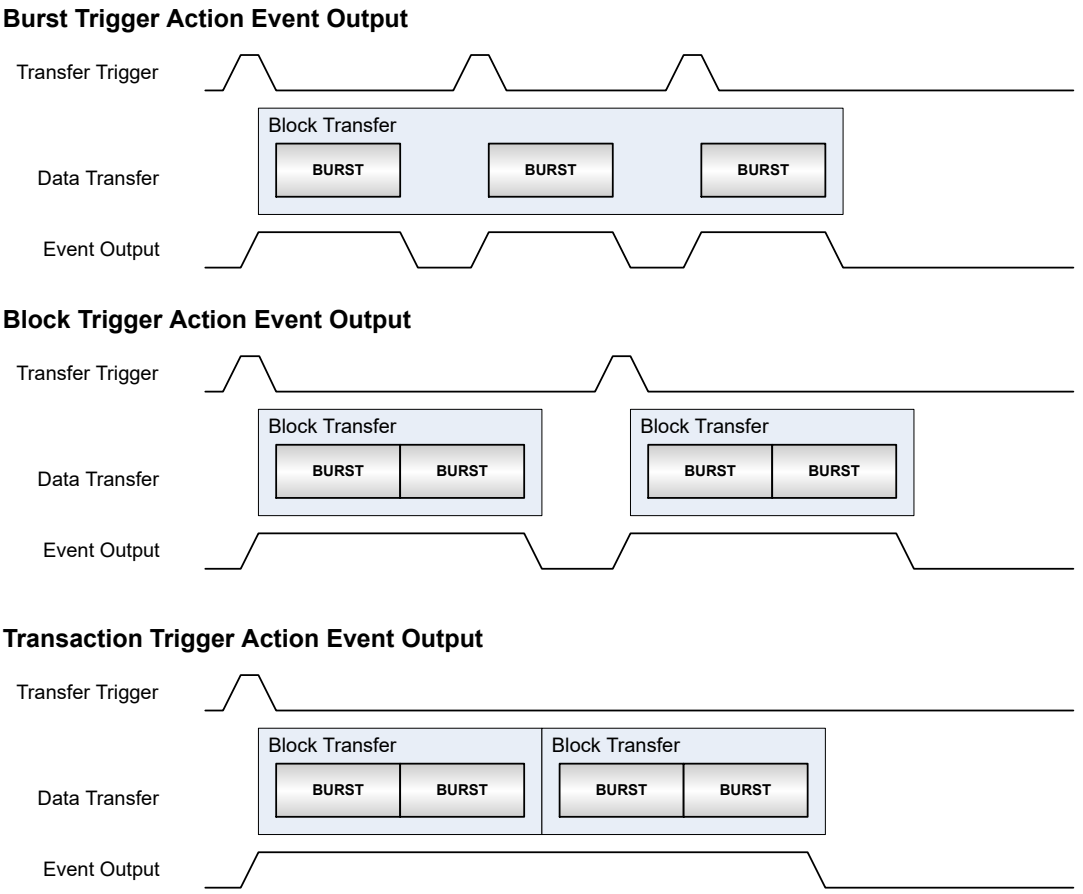



### Trigger action event output

When the trigger action event output is selected, an event level is generated. Then event output is set when the transfer trigger occurred, and cleared when the corresponding trigger action is completed. The figure below shows an example for each trigger action type.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-18.** Trigger Action Event Output Generation




### 22.6.3.7. Aborting Transfers

Transfers on any channel can be aborted gracefully by software by disabling the corresponding DMA channel. It is also possible to abort all ongoing or pending transfers by disabling the DMAC.

When a DMA channel disable request or DMAC disable request is detected:

- Ongoing transfers of the active channel will be disabled when the ongoing beat transfer is completed and the write-back memory section is updated. This prevents transfer corruption before the channel is disabled.
- All other enabled channels will be disabled in the next clock cycle.

The corresponding Channel Enable bit in the Channel Control A register is cleared (CHCTRLA.ENABLE=0) when the channel is disabled.

The corresponding DMAC Enable bit in the Control register is cleared (CTRL.DMAENABLE=0) when the entire DMAC module is disabled.

### 22.6.3.8. CRC Operation

A Cyclic Redundancy Check (CRC) is an error detection technique used to find errors in data. It is commonly used to determine whether the data during a transmission, or data present in data and program memories has been corrupted or not. A CRC takes a data stream or a block of data as input and generates a 16- or 32-bit output that can be appended to the data and used as a checksum.

When the data is received, the device or application repeats the calculation using the DSU's CRC engine. If the new CRC result does not match the one calculated earlier, the block contains a data error. The application will then detect this and may take a corrective action, such as requesting the data to be sent again or simply not using the incorrect data.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The CRC engine in DMAC supports two commonly used CRC polynomials: CRC-16 (CRC-CCITT) and CRC-32 (IEEE 802.3). Typically, applying CRC-n (CRC-16 or CRC-32) to a data block of arbitrary length will detect any single alteration that is ≤n bits in length, and will detect the fraction 1-2-n of all longer error bursts.

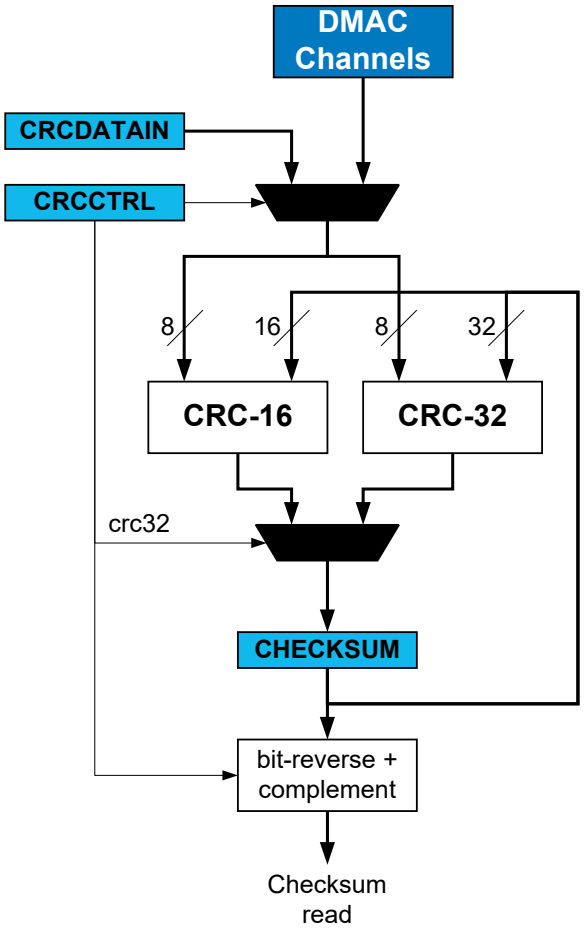
- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

The data source for the CRC engine can either be one of the DMA channels or the APB bus interface, and must be selected by writing to the CRC Input Source bits in the CRC Control register (CRCCTRL.CRCSRC). The CRC engine then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CRC Checksum register (CRCCHKSUM). When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented, as shown in *CRC Generator Block Diagram*.

The CRC polynomial is selected by writing to the CRC Polynomial Type bit in the CRC Control register (CRCCTRL.CRCPOLY), the default is CRC-16. The CRC engine operates on byte only. When the DMA is used as data source for the CRC engine, the DMA channel beat size setting will be used. When used with APB bus interface, the application must select the CRC Beat Size bit field of CRC Control register (CRCCTRL.CRCBEATSIZE). 8-, 16-, or 32-bit bus transfer access type is supported. The corresponding number of bytes will be written in the CRCDATAIN register and the CRC engine will operate on the input data in a byte by byte manner.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-19.** CRC Generator Block Diagram




| | |
|---|---|
| **CRC on DMA data** | CRC-16 or CRC-32 calculations can be performed on data passing through any DMA channel. Once a DMA channel is selected as the source, the CRC engine will continuously generate the CRC on the data passing through the DMA channel. The checksum is available for readout once the DMA transaction is completed or aborted. A CRC can also be generated on SRAM, Flash, or I/O memory by passing these data through a DMA channel. If the latter is done, the destination register for the DMA data can be the data input (CRCDATAIN) register in the CRC engine. |
| **CRC using the I/O interface** | Before using the CRC engine with the I/O interface, the application must set the CRC Beat Size bits in the CRC Control register (CRCCTRL.CRCBEATSIZE). 8/16/32-bit bus transfer type can be selected. |

CRC can be performed on any data by loading them into the CRC engine using the CPU and writing the data to the CRCDATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. This means if a 32-bit data is written to the CRCDATAIN register the CRC engine takes four cycles to calculate the CRC. The CRC complete is signaled by a set CRCBUSY bit in the CRCSTATUS register. New data can be written only when CRCBUSY flag is not set.

### 22.6.3.9. Memory CRC Generation

When enabled, it is possible to automatically calculate a memory block checksum. When the channel is enabled and the descriptor is fetched, the CRC Checksum register (CRCCHKSUM) is reloaded with the initial checksum value (CHKINIT) stored in the Block Transfer Destination Address register (DSTADDR). The DMA read and calculate the checksum over the data from the source address.When the checksum calculation is completed, the CRC value is stored in the CRC Checksum register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

(CRCCHKSUM), the Transfer Complete interrupt flag is set (CHINTFLAGn.TCMPL) and optional interrupt is generated.

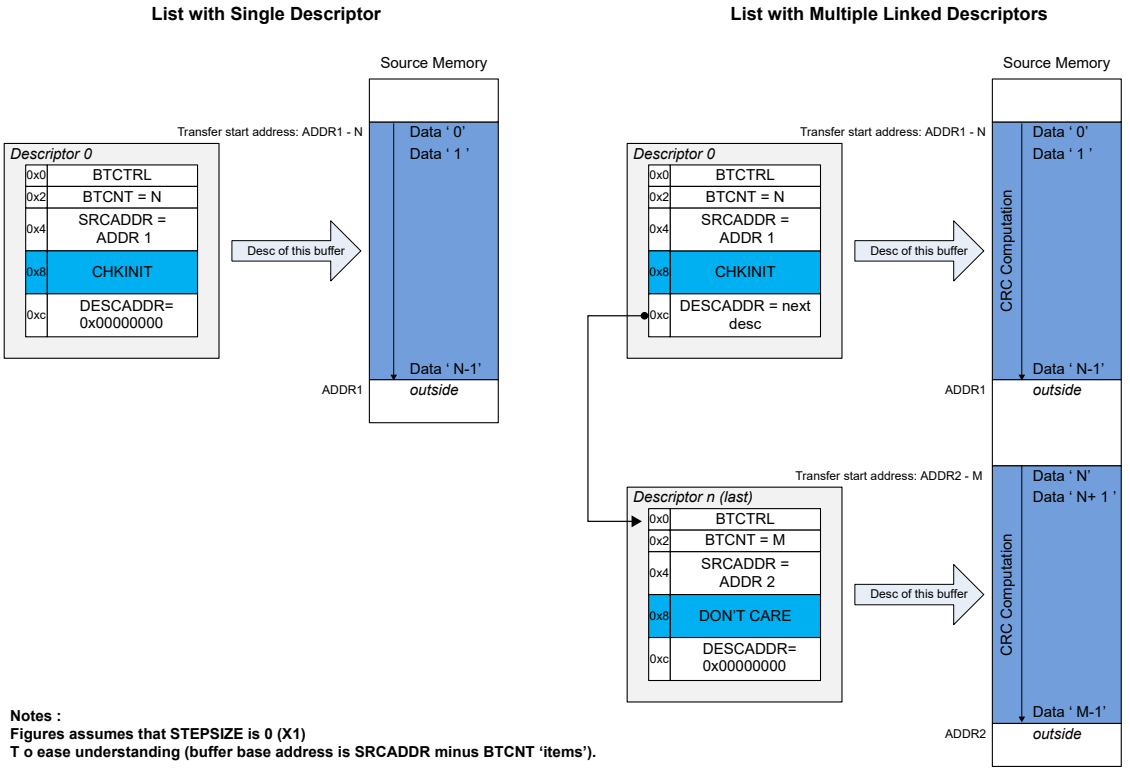
If linked descriptor is in the list (DESCADDR !=0), the DMA will fetch the next descriptor and CRC calculation continues as described above. When the last list descriptor is executed, the channel is automatically disabled.

In order to enable the memory CRC generation, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC)
2. Reserve memory space addresses to configure a descriptor or a list of descriptors
3. Configure each descriptor:
   - Set the next descriptor address (DESCADDR)
   - Set the destination address with the initial checksum value (DSTADDR = CHKINIT) in the first descriptior in a list
   - Set the transfer source address (SRCADDR)
   - Set the block transfer count (BTCNT)
   - Set the memory CRC generation operation mode (CRCCTRL.CRCMODE = CRCGEN)
   - Enable optional interrupts
4. Enable the corresponding DMA channel (CHCTRLAn.ENABLE)

The figure below shows the CRC computation slots and descriptor configuration when single or linked-descriptors transfers are enabled.

**Figure 22-20.** CRC Computation with Single Linked Transfers





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.6.3.10. Memory CRC Monitor

When enabled, it is possible to continuously check a a memory block data integrity by calculating and checking the CRC checksum. The expected CRC checksum value must be located in the last memory block location, as shown in the table below:

| CRCCTRL.CRCPOLY | CRCCTRL.CRCBEATSIZE | Last Memory Block Byte Locations Value (MSB Byte First) | CHECKSUM Result |
|---|---|---|---|
| CRC-16 | Byte | Expected CRC[7:0] | 0x00000000 |
| | Half-word | Expected CRC[15:8] | |
| | Word | 0x00<br>0x00<br>Expected CRC[7:0]<br>Expected CRC[15:8] | |
| CRC-32 | Byte | Expected CRC[31:24] | CRC Magic Number (0x2144DF1C) |
| | Half-word | Expected CRC[23:16] | |
| | Word | Expected CRC[15:8]<br>Expected CRC[7:0] | |

When the channel is enabled and the descriptor is fetched, the CRC Checksum register (CRCCHKSUM) is reloaded with the initial checksum value (CHKINIT), stored in the DSTADDR location of the first descriptor. The DMA read and calculate the checksum over the entire data from the source address.When the checksum calculation is completed the DMA read the last beat from the memory, the calculated CRC value from the CRC Checksum register is compared to zero or CRC magic number, depending on CRC polynomial selection.

If the CHECKSUM does not match the comparison value the DMA channel is disabled, and both and the CRC Error bit in the Channel n Status register (CHSTATUSn.CRCERR) and Transfer Error interrupt flag (CHINTFLAGn.TERR) are set. If enabled, the Transfer Error interrupt is generated.

If the calculated checksum value matches the compare value, the Transfer Complete interrupt flag (CHINTFLAGn.TCMPL) is set, optional interrupt is generated and the DMA will perform the following actions, depending on the descriptor list settings:

- If the list has only one descriptor, the DMA will re-fetch the descriptor
- If the current descriptor is the last descriptor from the list, the DMA will fetch the first descriptor from the list

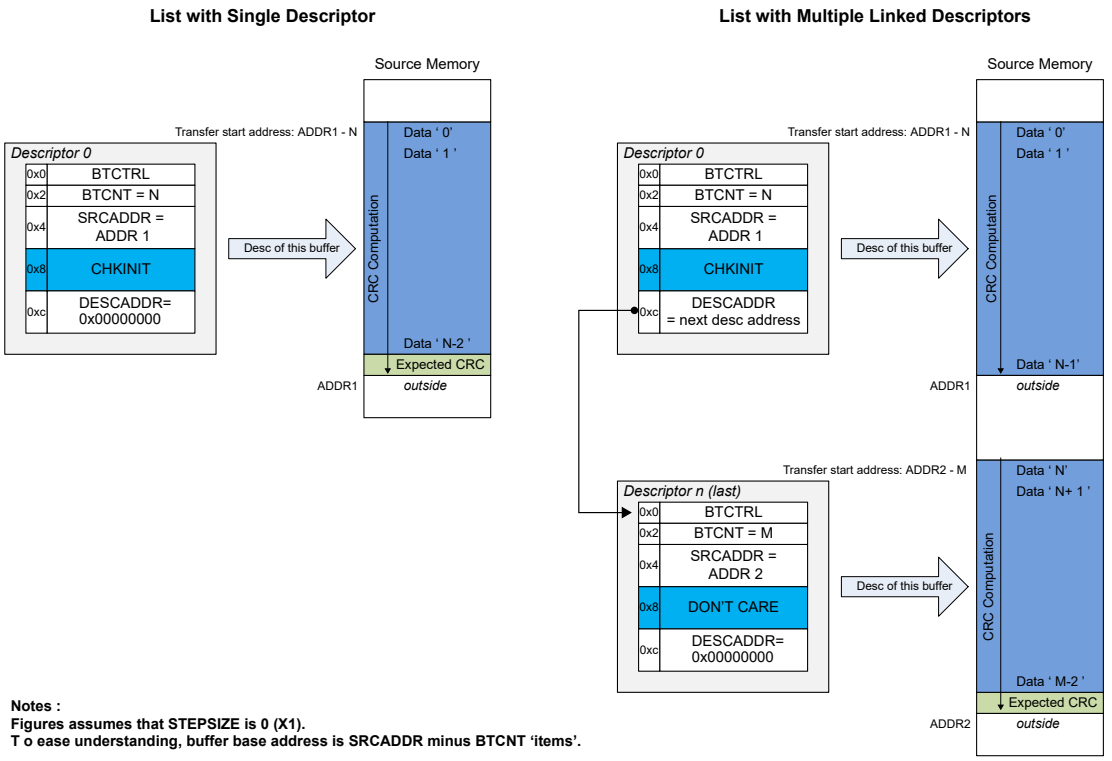
When the fetch is completed, the DMA restarts the operations described above when new triggers are detected.

In order to enable the memory CRC monitor, the following actions must be performed:

1. The CRC module must be set to be used with a DMA channel (CRCCTRL.CRCSRC)
2. Reserve memory space addresses to configure a descriptor or a list of descriptors
3. Configure each descriptor
   - Set the next descriptor address (DESCADDR)
   - In the first list descriptor, set the destination address with the initial checksum value (DSTADDR = CHKINIT)
   - Set the transfer source address (SRCADDR)
   - Set the block transfer count (BTCNT)
   - Set the memory CRC monitor operation mode (CRCCTRL.CRCMODE = CRCMON)
   - Enable optional interrupts
4. Enable the corresponding DMA channel (CHCTRLAn.ENABLE)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 22-21.** CRC Computation and Check with Single or Linked Transfers




**Notes :**
**Figures assumes that STEPSIZE is 0 (X1).**
**T o ease understanding, buffer base address is SRCADDR minus BTCNT 'items'.**

## 22.6.4. DMA Operation

Not applicable.

## 22.6.5. Interrupts

The DMAC channels have the following interrupt sources:

- Transfer Complete (TCMPL): Indicates that a block transfer is completed on the corresponding channel. Refer to Data Transmission for details.
- Transfer Error (TERR): Indicates that a bus error has occurred during a burst transfer, or that an invalid descriptor has been fetched. Refer to Error Handling for details.
- Channel Suspend (SUSP): Indicates that the corresponding channel has been suspended. Refer to Channel Suspend and Data Transmission for details.

Each interrupt source has an Interrupt flag associated with it. The Interrupt flag in the Channel Interrupt Flag Status and Clear (CHINTFLAG) register is set when the Interrupt condition occurs. Each interrupt can be individually enabled by setting the corresponding bit in the Channel Interrupt Enable Set register (CHINTENSET=1), and disabled by setting the corresponding bit in the Channel Interrupt Enable Clear register (CHINTENCLR=1).

An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the Interrupt flag is cleared, the interrupt is disabled, the DMAC is reset or the corresponding DMA channel is reset. See CHINTFLAG for details on how to clear Interrupt flags. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts and must read the Channel Interrupt Flag Status and Clear (CHINTFLAG) register to determine which Interrupt condition is present for the corresponding channel. It is also possible


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective Interrupt flags.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

### 22.6.6. Events

The DMAC can generate the following output events:

- Channel (CH): Generated when a block transfer for a given channel has been completed, or when a beat transfer within a block transfer for a given channel has been completed. Refer to *Event Output Selection* for details.

Setting the Channel Event Output Enable bit (CHEVCTRLx.EVOE = 1) enables the corresponding output event configured in the Event Output Selection bit group in the Block Transfer Control register (BTCTRL.EVOSEL). Clearing CHEVCTRLx.EVOE = 0 disables the corresponding output event.

The DMAC can take the following actions on an input event:

- Transfer and Periodic Transfer Trigger (TRIG): normal transfer or periodic transfers on peripherals are enabled
- Conditional Transfer Trigger (CTRIG): conditional transfers on peripherals are enabled
- Conditional Block Transfer Trigger (CBLOCK): conditional block transfers on peripherals are enabled
- Channel Suspend Operation (SUSPEND): suspend a channel operation
- Channel Resume Operation (RESUME): resume a suspended channel operation
- Skip Next Block Suspend Action (SSKIP): skip the next block suspend transfer condition
- Increase Priority (INCPRI): increase channel priority

Setting the Channel Event Input Enable bit (CHEVCTRLx.EVIE = 1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event. Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. For further details on event input actions, refer to *Event Input Actions*.
**Note:** Event input and outputs are not available for every channel. Refer to the Features section for more information.

**Related Links**
Event System (EVSYS)
Event Output Selection
Event Input Actions

### 22.6.7. Sleep Mode Operation

### 22.6.8. Synchronization

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 22.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRL | 7:0 | | | | | | | DMAENABLE | SWRST |
| | | 15:8 | | | | | LVLEN3 | LVLEN2 | LVLEN1 | LVLEN0 |
| 0x02 | CRCCTRL | 7:0 | | | | | CRCPOLY[1:0] | | CRCBEATSIZE[1:0] | |
| | | 15:8 | CRCMODE[1:0] | | CRCSRC[5:0] | | | | | |
| 0x04 | CRCDATAIN | 7:0 | CRCDATAIN[7:0] | | | | | | | |
| | | 15:8 | CRCDATAIN[15:8] | | | | | | | |
| | | 23:16 | CRCDATAIN[23:16] | | | | | | | |
| | | 31:24 | CRCDATAIN[31:24] | | | | | | | |
| 0x08 | CRCCHKSUM | 7:0 | CRCCHKSUM[7:0] | | | | | | | |
| | | 15:8 | CRCCHKSUM[15:8] | | | | | | | |
| | | 23:16 | CRCCHKSUM[23:16] | | | | | | | |
| | | 31:24 | CRCCHKSUM[31:24] | | | | | | | |
| 0x0C | CRCSTATUS | 7:0 | | | | | | CRCERR | CRCZERO | CRCBUSY |
| 0x0D | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0E ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | SWTRIGCTRL | 7:0 | SWTRIG[7:0] | | | | | | | |
| | | 15:8 | SWTRIG[15:8] | | | | | | | |
| | | 23:16 | SWTRIG[23:16] | | | | | | | |
| | | 31:24 | SWTRIG[31:24] | | | | | | | |
| 0x14 | PRICTRL0 | 7:0 | RRLVLEN0 | QOS0[1:0] | | LVLPRI0[4:0] | | | | |
| | | 15:8 | RRLVLEN1 | QOS1[1:0] | | LVLPRI1[4:0] | | | | |
| | | 23:16 | RRLVLEN2 | QOS2[1:0] | | LVLPRI2[4:0] | | | | |
| | | 31:24 | RRLVLEN3 | QOS3[1:0] | | LVLPRI3[4:0] | | | | |
| 0x18 ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | INTPEND | 7:0 | | | | ID[4:0] | | | | |
| | | 15:8 | PEND | BUSY | FERR | CRCERR | | SUSP | TCMPL | TERR |
| 0x22 ... 0x23 | Reserved | | | | | | | | | |
| 0x24 | INTSTATUS | 7:0 | CHINT7 | CHINT6 | CHINT5 | CHINT4 | CHINT3 | CHINT2 | CHINT1 | CHINT0 |
| | | 15:8 | CHINT15 | CHINT14 | CHINT13 | CHINT12 | CHINT11 | CHINT10 | CHINT9 | CHINT8 |
| | | 23:16 | CHINT23 | CHINT22 | CHINT21 | CHINT20 | CHINT19 | CHINT18 | CHINT17 | CHINT16 |
| | | 31:24 | CHINT31 | CHINT30 | CHINT29 | CHINT28 | CHINT27 | CHINT26 | CHINT25 | CHINT24 |
| 0x28 | BUSYCH | 7:0 | BUSYCH7 | BUSYCH6 | BUSYCH5 | BUSYCH4 | BUSYCH3 | BUSYCH2 | BUSYCH1 | BUSYCH0 |
| | | 15:8 | BUSYCH15 | BUSYCH14 | BUSYCH13 | BUSYCH12 | BUSYCH11 | BUSYCH10 | BUSYCH9 | BUSYCH8 |
| | | 23:16 | BUSYCH23 | BUSYCH22 | BUSYCH21 | BUSYCH20 | BUSYCH19 | BUSYCH18 | BUSYCH17 | BUSYCH16 |
| | | 31:24 | BUSYCH31 | BUSYCH30 | BUSYCH29 | BUSYCH28 | BUSYCH27 | BUSYCH26 | BUSYCH25 | BUSYCH24 |
| 0x2C | PENDCH | 7:0 | PENDCH7 | PENDCH6 | PENDCH5 | PENDCH4 | PENDCH3 | PENDCH2 | PENDCH1 | PENDCH0 |
| | | 15:8 | PENDCH15 | PENDCH14 | PENDCH13 | PENDCH12 | PENDCH11 | PENDCH10 | PENDCH9 | PENDCH8 |
| | | 23:16 | PENDCH23 | PENDCH22 | PENDCH21 | PENDCH20 | PENDCH19 | PENDCH18 | PENDCH17 | PENDCH16 |
| | | 31:24 | PENDCH31 | PENDCH30 | PENDCH29 | PENDCH28 | PENDCH27 | PENDCH26 | PENDCH25 | PENDCH24 |
| 0x30 | ACTIVE | 7:0 | | | | | LVLEX3 | LVLEX2 | LVLEX1 | LVLEX0 |
| | | 15:8 | ABUSY | | | ID[4:0] | | | | |
| | | 23:16 | BTCNT[7:0] | | | | | | | |
| | | 31:24 | BTCNT[15:8] | | | | | | | |
| 0x34 | BASEADDR | 7:0 | BASEADDR[7:0] | | | | | | | |
| | | 15:8 | BASEADDR[15:8] | | | | | | | |
| | | 23:16 | BASEADDR[23:16] | | | | | | | |
| | | 31:24 | BASEADDR[31:24] | | | | | | | |
| 0x38 | WRBADDR | 7:0 | WRBADDR[7:0] | | | | | | | |
| | | 15:8 | WRBADDR[15:8] | | | | | | | |
| | | 23:16 | WRBADDR[23:16] | | | | | | | |
| | | 31:24 | WRBADDR[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x3C ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | CHCTRLA0 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x44 | CHCTRLB0 | 7:0 | | | | | | | CMD[1:0] | |
| 0x45 | CHPRILVL0 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x46 | CHEVCTRL0 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x47 ... 0x4B | Reserved | | | | | | | | | |
| 0x4C | CHINTENCLR0 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x4D | CHINTENSET0 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x4E | CHINTFLAG0 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x4F | CHSTATUS0 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x50 | CHCTRLA1 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x54 | CHCTRLB1 | 7:0 | | | | | | | CMD[1:0] | |
| 0x55 | CHPRILVL1 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x56 | CHEVCTRL1 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x57 ... 0x5B | Reserved | | | | | | | | | |
| 0x5C | CHINTENCLR1 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x5D | CHINTENSET1 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x5E | CHINTFLAG1 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x5F | CHSTATUS1 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x60 | CHCTRLA2 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x64 | CHCTRLB2 | 7:0 | | | | | | | CMD[1:0] | |
| 0x65 | CHPRILVL2 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x66 | CHEVCTRL2 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x67 ... 0x6B | Reserved | | | | | | | | | |
| 0x6C | CHINTENCLR2 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x6D | CHINTENSET2 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x6E | CHINTFLAG2 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x6F | CHSTATUS2 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x70 | CHCTRLA3 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x74 | CHCTRLB3 | 7:0 | | | | | | | CMD[1:0] | |
| 0x75 | CHPRILVL3 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x76 | CHEVCTRL3 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x77 ... 0x7B | Reserved | | | | | | | | | |
| 0x7C | CHINTENCLR3 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x7D | CHINTENSET3 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x7E | CHINTFLAG3 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x7F | CHSTATUS3 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x80 | CHCTRLA4 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x84 | CHCTRLB4 | 7:0 | | | | | | | CMD[1:0] | |
| 0x85 | CHPRILVL4 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x86 | CHEVCTRL4 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x87 ... 0x8B | Reserved | | | | | | | | | |
| 0x8C | CHINTENCLR4 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x8D | CHINTENSET4 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x8E | CHINTFLAG4 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x8F | CHSTATUS4 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x90 | CHCTRLA5 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x94 | CHCTRLB5 | 7:0 | | | | | | | CMD[1:0] | |
| 0x95 | CHPRILVL5 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x96 | CHEVCTRL5 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x97 ... 0x9B | Reserved | | | | | | | | | |
| 0x9C | CHINTENCLR5 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x9D | CHINTENSET5 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x9E | CHINTFLAG5 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x9F | CHSTATUS5 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0xA0 | CHCTRLA6 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0xA4 | CHCTRLB6 | 7:0 | | | | | | | CMD[1:0] | |
| 0xA5 | CHPRILVL6 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0xA6 | CHEVCTRL6 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0xA7 ... 0xAB | Reserved | | | | | | | | | |
| 0xAC | CHINTENCLR6 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xAD | CHINTENSET6 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xAE | CHINTFLAG6 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xAF | CHSTATUS6 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0xB0 | CHCTRLA7 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0xB4 | CHCTRLB7 | 7:0 | | | | | | | CMD[1:0] | |
| 0xB5 | CHPRILVL7 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0xB6 | CHEVCTRL7 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0xB7 ... 0xBB | Reserved | | | | | | | | | |
| 0xBC | CHINTENCLR7 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xBD | CHINTENSET7 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xBE | CHINTFLAG7 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xBF | CHSTATUS7 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0xC0 | CHCTRLA8 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0xC4 | CHCTRLB8 | 7:0 | | | | | | | CMD[1:0] | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xC5 | CHPRILVL8 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0xC6 | CHEVCTRL8 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0xC7 ... 0xCB | Reserved | | | | | | | | | |
| 0xCC | CHINTENCLR8 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xCD | CHINTENSET8 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xCE | CHINTFLAG8 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xCF | CHSTATUS8 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0xD0 | CHCTRLA9 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0xD4 | CHCTRLB9 | 7:0 | | | | | | | CMD[1:0] | |
| 0xD5 | CHPRILVL9 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0xD6 | CHEVCTRL9 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0xD7 ... 0xDB | Reserved | | | | | | | | | |
| 0xDC | CHINTENCLR9 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xDD | CHINTENSET9 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xDE | CHINTFLAG9 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xDF | CHSTATUS9 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0xE0 | CHCTRLA10 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0xE4 | CHCTRLB10 | 7:0 | | | | | | | CMD[1:0] | |
| 0xE5 | CHPRILVL10 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0xE6 | CHEVCTRL10 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0xE7 ... 0xEB | Reserved | | | | | | | | | |
| 0xEC | CHINTENCLR10 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xED | CHINTENSET10 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xEE | CHINTFLAG10 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xEF | CHSTATUS10 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0xF0 | CHCTRLA11 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0xF4 | CHCTRLB11 | 7:0 | | | | | | | CMD[1:0] | |
| 0xF5 | CHPRILVL11 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0xF6 | CHEVCTRL11 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0xF7 ... 0xFB | Reserved | | | | | | | | | |
| 0xFC | CHINTENCLR11 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xFD | CHINTENSET11 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xFE | CHINTFLAG11 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0xFF | CHSTATUS11 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0100 | CHCTRLA12 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0104 | CHCTRLB12 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0105 | CHPRILVL12 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0106 | CHEVCTRL12 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0107 ... 0x010B | Reserved | | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x010C | CHINTENCLR12 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x010D | CHINTENSET12 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x010E | CHINTFLAG12 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x010F | CHSTATUS12 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0110 | CHCTRLA13 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0114 | CHCTRLB13 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0115 | CHPRILVL13 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0116 | CHEVCTRL13 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0117 ... 0x011B | Reserved | | | | | | | | | |
| 0x011C | CHINTENCLR13 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x011D | CHINTENSET13 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x011E | CHINTFLAG13 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x011F | CHSTATUS13 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0120 | CHCTRLA14 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0124 | CHCTRLB14 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0125 | CHPRILVL14 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0126 | CHEVCTRL14 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0127 ... 0x012B | Reserved | | | | | | | | | |
| 0x012C | CHINTENCLR14 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x012D | CHINTENSET14 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x012E | CHINTFLAG14 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x012F | CHSTATUS14 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0130 | CHCTRLA15 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0134 | CHCTRLB15 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0135 | CHPRILVL15 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0136 | CHEVCTRL15 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0137 ... 0x013B | Reserved | | | | | | | | | |
| 0x013C | CHINTENCLR15 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x013D | CHINTENSET15 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x013E | CHINTFLAG15 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x013F | CHSTATUS15 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0140 | CHCTRLA16 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0144 | CHCTRLB16 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0145 | CHPRILVL16 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0146 | CHEVCTRL16 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0147 ... 0x014B | Reserved | | | | | | | | | |
| 0x014C | CHINTENCLR16 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x014D | CHINTENSET16 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x014E | CHINTFLAG16 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x014F | CHSTATUS16 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0150 | CHCTRLA17 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0154 | CHCTRLB17 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0155 | CHPRILVL17 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0156 | CHEVCTRL17 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0157 ... 0x015B | Reserved | | | | | | | | | |
| 0x015C | CHINTENCLR17 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x015D | CHINTENSET17 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x015E | CHINTFLAG17 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x015F | CHSTATUS17 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0160 | CHCTRLA18 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0164 | CHCTRLB18 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0165 | CHPRILVL18 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0166 | CHEVCTRL18 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0167 ... 0x016B | Reserved | | | | | | | | | |
| 0x016C | CHINTENCLR18 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x016D | CHINTENSET18 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x016E | CHINTFLAG18 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x016F | CHSTATUS18 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0170 | CHCTRLA19 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0174 | CHCTRLB19 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0175 | CHPRILVL19 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0176 | CHEVCTRL19 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0177 ... 0x017B | Reserved | | | | | | | | | |
| 0x017C | CHINTENCLR19 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x017D | CHINTENSET19 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x017E | CHINTFLAG19 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x017F | CHSTATUS19 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0180 | CHCTRLA20 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0184 | CHCTRLB20 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0185 | CHPRILVL20 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0186 | CHEVCTRL20 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0187 ... 0x018B | Reserved | | | | | | | | | |
| 0x018C | CHINTENCLR20 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x018D | CHINTENSET20 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x018E | CHINTFLAG20 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x018F | CHSTATUS20 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0190 | CHCTRLA21 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0194 | CHCTRLB21 | 7:0 | | | | | | | CMD[1:0] | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0195 | CHPRILVL21 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0196 | CHEVCTRL21 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0197 ... 0x019B | Reserved | | | | | | | | | |
| 0x019C | CHINTENCLR21 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x019D | CHINTENSET21 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x019E | CHINTFLAG21 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x019F | CHSTATUS21 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x01A0 | CHCTRLA22 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x01A4 | CHCTRLB22 | 7:0 | | | | | | | CMD[1:0] | |
| 0x01A5 | CHPRILVL22 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x01A6 | CHEVCTRL22 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x01A7 ... 0x01AB | Reserved | | | | | | | | | |
| 0x01AC | CHINTENCLR22 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01AD | CHINTENSET22 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01AE | CHINTFLAG22 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01AF | CHSTATUS22 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x01B0 | CHCTRLA23 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x01B4 | CHCTRLB23 | 7:0 | | | | | | | CMD[1:0] | |
| 0x01B5 | CHPRILVL23 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x01B6 | CHEVCTRL23 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x01B7 ... 0x01BB | Reserved | | | | | | | | | |
| 0x01BC | CHINTENCLR23 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01BD | CHINTENSET23 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01BE | CHINTFLAG23 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01BF | CHSTATUS23 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x01C0 | CHCTRLA24 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x01C4 | CHCTRLB24 | 7:0 | | | | | | | CMD[1:0] | |
| 0x01C5 | CHPRILVL24 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x01C6 | CHEVCTRL24 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x01C7 ... 0x01CB | Reserved | | | | | | | | | |
| 0x01CC | CHINTENCLR24 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01CD | CHINTENSET24 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01CE | CHINTFLAG24 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01CF | CHSTATUS24 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x01D0 | CHCTRLA25 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x01D4 | CHCTRLB25 | 7:0 | | | | | | | CMD[1:0] | |
| 0x01D5 | CHPRILVL25 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x01D6 | CHEVCTRL25 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x01D7 ... 0x01DB | Reserved | | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01DC | CHINTENCLR25 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01DD | CHINTENSET25 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01DE | CHINTFLAG25 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01DF | CHSTATUS25 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x01E0 | CHCTRLA26 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x01E4 | CHCTRLB26 | 7:0 | | | | | | | CMD[1:0] | |
| 0x01E5 | CHPRILVL26 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x01E6 | CHEVCTRL26 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x01E7 ... 0x01EB | Reserved | | | | | | | | | |
| 0x01EC | CHINTENCLR26 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01ED | CHINTENSET26 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01EE | CHINTFLAG26 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01EF | CHSTATUS26 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x01F0 | CHCTRLA27 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x01F4 | CHCTRLB27 | 7:0 | | | | | | | CMD[1:0] | |
| 0x01F5 | CHPRILVL27 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x01F6 | CHEVCTRL27 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x01F7 ... 0x01FB | Reserved | | | | | | | | | |
| 0x01FC | CHINTENCLR27 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01FD | CHINTENSET27 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01FE | CHINTFLAG27 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x01FF | CHSTATUS27 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0200 | CHCTRLA28 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0204 | CHCTRLB28 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0205 | CHPRILVL28 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0206 | CHEVCTRL28 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0207 ... 0x020B | Reserved | | | | | | | | | |
| 0x020C | CHINTENCLR28 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x020D | CHINTENSET28 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x020E | CHINTFLAG28 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x020F | CHSTATUS28 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0210 | CHCTRLA29 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0214 | CHCTRLB29 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0215 | CHPRILVL29 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0216 | CHEVCTRL29 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0217 ... 0x021B | Reserved | | | | | | | | | |
| 0x021C | CHINTENCLR29 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x021D | CHINTENSET29 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x021E | CHINTFLAG29 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x021F | CHSTATUS29 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0220 | CHCTRLA30 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0224 | CHCTRLB30 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0225 | CHPRILVL30 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0226 | CHEVCTRL30 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0227 ... 0x022B | Reserved | | | | | | | | | |
| 0x022C | CHINTENCLR30 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x022D | CHINTENSET30 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x022E | CHINTFLAG30 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x022F | CHSTATUS30 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |
| 0x0230 | CHCTRLA31 | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| | | 15:8 | | TRIGSRC[6:0] | | | | | | |
| | | 23:16 | | | TRIGACT[1:0] | | | | | |
| | | 31:24 | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| 0x0234 | CHCTRLB31 | 7:0 | | | | | | | CMD[1:0] | |
| 0x0235 | CHPRILVL31 | 7:0 | | | | | | | PRILVL[1:0] | |
| 0x0236 | CHEVCTRL31 | 7:0 | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| 0x0237 ... 0x023B | Reserved | | | | | | | | | |
| 0x023C | CHINTENCLR31 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x023D | CHINTENSET31 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x023E | CHINTFLAG31 | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x023F | CHSTATUS31 | 7:0 | | | | | CRCERR | FERR | BUSY | PEND |

## 22.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.1. Control

**Name:** CTRL
**Offset:** 0x00
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LVLEN3 | LVLEN2 | LVLEN1 | LVLEN0 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DMAENABLE | SWRST |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 8, 9, 10, 11 – LVLENx** Priority Level x Enable
When this bit is set, all requests with the corresponding level will be fed into the arbiter block. When cleared, all requests with the corresponding level will be ignored.
For details on arbitration schemes, refer to the Arbitration section.
These bits are not enable-protected.

| Value | Description |
|---|---|
| 0 | Transfer requests for Priority level x will not be handled. |
| 1 | Transfer requests for Priority level x will be handled. |

**Bit 1 – DMAENABLE** DMA Enable
Setting this bit will enable the DMA module.
Writing a '0' to this bit will disable the DMA module. When writing a '0' during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit when the DMAC module is disabled (DMAENABLE bit set to "0"), resets all registers in the DMAC (except DBGCTRL) to their initial state. If either the DMAC or CRC module is enabled, the Reset request will be ignored and the DMAC will return an access error.

| Value | Description |
|---|---|
| 0 | There is no Reset operation ongoing. |
| 1 | A Reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.2. CRC Control

**Name:** CRCCTRL
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CRCMODE[1:0] | | CRCSRC[5:0] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CRCPOLY[1:0] | | CRCBEATSIZE[1:0] | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 15:14 – CRCMODE[1:0]** CRC Operating Mode
These bits define the block transfer mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Default operating mode |
| 0x1 | - | Reserved |
| 0x2 | CRCMON | Memory CRC monitor operating mode |
| 0x3 | CRCGEN | Memory CRC generation operating mode |

**Bits 13:8 – CRCSRC[5:0]** CRC Input Source
These bits select the input source for generating the CRC. The selected source is locked until either the CRC generation is completed or the CRC module is disabled. This means the CRCSRC cannot be modified when the CRC operation is ongoing. The lock is signaled by the CRCBUSY status bit. CRC generation complete is generated and signaled from the selected source when used with the DMA channel.

| Value | Name | Description |
|---|---|---|
| 0x00 | DISABLE | No action |
| 0x01 | IO | I/O interface |
| 0x02 – 0x1F | - | Reserved |
| 0x20 | CH0 | DMA channel 0 |
| 0x21 | CH1 | DMA channel 1 |
| 0x22 | CH2 | DMA channel 2 |
| 0x23 | CH3 | DMA channel 3 |
| 0x24 | CH4 | DMA channel 4 |
| 0x25 | CH5 | DMA channel 5 |
| 0x26 | CH6 | DMA channel 6 |
| 0x27 | CH7 | DMA channel 7 |
| 0x28 | CH8 | DMA channel 8 |
| 0x29 | CH9 | DMA channel 9 |
| 0x2A | CH10 | DMA channel 10 |
| 0x2B | CH11 | DMA channel 11 |
| 0x2C | CH12 | DMA channel 12 |
| 0x2D | CH13 | DMA channel 13 |
| 0x2E | CH14 | DMA channel 14 |
| 0x2F | CH15 | DMA channel 15 |
| 0x30 | CH16 | DMA channel 16 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x31 | CH17 | DMA channel 17 |
| 0x32 | CH18 | DMA channel 18 |
| 0x33 | CH19 | DMA channel 19 |
| 0x34 | CH20 | DMA channel 20 |
| 0x35 | CH21 | DMA channel 21 |
| 0x36 | CH22 | DMA channel 22 |
| 0x37 | CH23 | DMA channel 23 |
| 0x38 | CH24 | DMA channel 24 |
| 0x39 | CH25 | DMA channel 25 |
| 0x3A | CH26 | DMA channel 26 |
| 0x3B | CH27 | DMA channel 27 |
| 0x3C | CH28 | DMA channel 28 |
| 0x3D | CH29 | DMA channel 29 |
| 0x3E | CH30 | DMA channel 30 |
| 0x3F | CH31 | DMA channel 31 |

**Bits 3:2 – CRCPOLY[1:0]** CRC Polynomial Type
These bits select the CRC polynomial type.

| Value | Name | Description |
|---|---|---|
| 0x0 | CRC16 | CRC-16 (CRC-CCITT) |
| 0x1 | CRC32 | CRC32 (IEEE 802.3) |
| 0x2-0x3 | - | Reserved |

**Bits 1:0 – CRCBEATSIZE[1:0]** CRC Beat Size
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface.

| Value | Name | Description |
|---|---|---|
| 0x0 | BYTE | 8-bit bus transfer |
| 0x1 | HWORD | 16-bit bus transfer |
| 0x2 | WORD | 32-bit bus transfer |
| 0x3 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.3. CRC Data Input

**Name:** CRCDATAIN
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRCDATAIN[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CRCDATAIN[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CRCDATAIN[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CRCDATAIN[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CRCDATAIN[31:0]** CRC Data Input

These bits store the data for which the CRC checksum is computed. A new CRC checksum is ready (CRCBEAT+ 1) clock cycles after the CRCDATAIN register is written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.4. CRC Checksum

**Name:** CRCCHKSUM
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write Protection, Enable-Protected

The CRCCHKSUM represents the 16- or 32-bit checksum value and the generated CRC. The register is reset to zero by default, but it is possible to reset all bits to one by writing the CRCCHKSUM register directly. It is possible to write this register only when the CRC module is disabled. If CRC-32 is selected and the CRC Status Busy flag is cleared (i.e., CRC generation is completed or aborted), the bit reversed (bit 31 is swapped with bit 0, bit 30 with bit 1, etc.) and complemented result will be read from CRCCHKSUM. If CRC-16 is selected or the CRC Status Busy flag is set (i.e., CRC generation is ongoing), CRCCHKSUM will contain the actual content.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRCCHKSUM[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CRCCHKSUM[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CRCCHKSUM[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CRCCHKSUM[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CRCCHKSUM[31:0]** CRC Checksum
These bits store the generated CRC result. The 16 MSB bits are always read zero when CRC-16 is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.5. CRC Status

**Name:** CRCSTATUS
**Offset:** 0x0C
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CRCERR | CRCZERO | CRCBUSY |
| Access | | | | | | R | R | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – CRCERR** CRC Error
This bit is read '1' when the memory CRC monitor detects data corruption.

**Bit 1 – CRCZERO** CRC Zero
This bit is cleared when a new CRC source is selected.
This bit is set when the CRC generation is complete and the CRC Checksum is zero.

**Bit 0 – CRCBUSY** CRC Module Busy
When used with an I/O interface (CRCCTRL.CRCSRC=0x1):

- This bit is cleared by writing a '1' to it
- This bit is set when the CRC Data Input (CRCDATAIN) register is written
- Writing a '1' to this bit will clear the CRC Module Busy bit
- Writing a '0' to this bit has no effect

When used with a DMA channel (CRCCTRL.CRCSRC=0x20..,0x3F):

- This bit is cleared when the corresponding DMA channel is disabled
- This bit is set when the corresponding DMA channel is enabled
- Writing a '1' to this bit has no effect
- Writing a '0' to this bit has no effect


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.6. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0D
**Reset:** 0x00
**Property:** PAC Write Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
This bit is not reset by a Software Reset.
This bit controls the functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The DMAC is halted when the CPU is halted by an external debugger. |
| 1 | The DMAC continues normal operation when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.7. Software Trigger Control

**Name:** SWTRIGCTRL
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SWTRIG[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SWTRIG[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SWTRIG[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SWTRIG[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – SWTRIG[31:0]** Channel n Software Trigger [n = 31..0]
This bit is cleared when the Channel Pending bit in the Channel Status register (CHSTATUS.PEND) for the corresponding channel is either set, or by writing a '1' to it.
This bit is set if CHSTATUS.PEND is already '1' when writing a '1' to that bit.
Writing a '0' to this bit will clear the bit.
Writing a '1' to this bit will generate a DMA software trigger on channel x, if CHSTATUS.PEND=0 for channel x. CHSTATUS.PEND will be set and SWTRIGn will remain cleared.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.8. Priority Control 0

**Name:** PRICTRL0
**Offset:** 0x14
**Reset:** 0x40404040
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN3 | QOS3[1:0] | | LVLPRI3[4:0] | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN2 | QOS2[1:0] | | LVLPRI2[4:0] | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN1 | QOS1[1:0] | | LVLPRI1[4:0] | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN0 | QOS0[1:0] | | LVLPRI0[4:0] | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7, 15, 23, 31 – RRLVLEN** Level Round-Robin Scheduling Enable
For details on arbitration schemes, refer to Arbitration.

| Value | Description |
|---|---|
| 0 | Static arbitration scheme for channels with level 0 priority. |
| 1 | Round-robin arbitration scheme for channels with level 0 priority. |

**Bits 5:6, 13:14, 21:22, 29:30 – QOS** Level Quality of Service

| Value | Description |
|---|---|
| 0x0 | Disable Background (no sensitive operation). |
| 0x1 | Low Sensitive to bandwidth. |
| 0x2 | Medium Sensitive to latency. |
| 0x3 | Critical Latency. |

**Bits 0:4, 8:12, 16:20, 24:28 – LVLPRI** Level Channel Priority Number
When round-robin arbitration is enabled (PRICTRL0.RRLVLEN0=1) for priority level 0, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 0.
When static arbitration is enabled (PRICTRL0.RRLVLEN0=0) for priority level 0, and the value of this bit group is non-zero, it will not affect the static priority scheme.
This bit group is not reset when round-robin arbitration gets disabled (PRICTRL0.RRLVLEN0 written to '0').


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.9. Interrupt Pending

**Name:** INTPEND
**Offset:** 0x20
**Reset:** 0x0000
**Property:** -

This register allows the user to identify the lowest DMA channel with pending interrupt.
An interrupt that handles several channels should consult the INTPEND register to find out which channel number has priority (ignoring/filtering each channel that has its own interrupt line). An interrupt dedicated to only one channel must not use the INTPEND register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PEND | BUSY | FERR | CRCERR | | SUSP | TCMPL | TERR |
| Access | R | R | R | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | ID[4:0] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – PEND** Pending
This bit will read '1' when the channel selected by Channel ID field (ID) is pending.

**Bit 14 – BUSY** Busy
This bit will read '1' when the channel selected by Channel ID field (ID) is busy.

**Bit 13 – FERR** Fetch Error
This bit will read '1' when the channel selected by Channel ID field (ID) fetched an invalid descriptor.

**Bit 12 – CRCERR** CRC Error
This bit will read '1' when the channel selected by Channel ID field (ID) has a CRC Error Status Flag bit set, and is set when the CRC monitor detects data corruption.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

**Bit 10 – SUSP** Channel Suspend
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Suspend interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

**Bit 9 – TCMPL** Transfer Complete
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Complete interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

**Bit 8 – TERR** Transfer Error
This bit will read '1' when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn), where n is determined by the Channel ID bit field (ID).

**Bits 4:0 – ID[4:0]** Channel ID

These bits store the lowest channel number with pending interrupts. The number is valid if Suspend (SUSP), Transfer Complete (TCMPL) or Transfer Error (TERR) bits are set. The Channel ID field is refreshed when a new channel (with channel number less than the current one) with pending interrupts is detected, or when the application clears the corresponding channel interrupt sources. When no pending channels interrupts are available, these bits will always return zero value when read.
When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.10. Interrupt Status

**Name:** INTSTATUS
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CHINT31 | CHINT30 | CHINT29 | CHINT28 | CHINT27 | CHINT26 | CHINT25 | CHINT24 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CHINT23 | CHINT22 | CHINT21 | CHINT20 | CHINT19 | CHINT18 | CHINT17 | CHINT16 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CHINT15 | CHINT14 | CHINT13 | CHINT12 | CHINT11 | CHINT10 | CHINT9 | CHINT8 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CHINT7 | CHINT6 | CHINT5 | CHINT4 | CHINT3 | CHINT2 | CHINT1 | CHINT0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CHINTn** Channel n Pending Interrupt [n=31..0]
This bit is set when Channel n has a pending interrupt/the interrupt request is received.
This bit is cleared when the corresponding Channel n interrupts are disabled or the interrupts sources are cleared.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.11. Busy Channels

**Name:** BUSYCH
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BUSYCH31 | BUSYCH30 | BUSYCH29 | BUSYCH28 | BUSYCH27 | BUSYCH26 | BUSYCH25 | BUSYCH24 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BUSYCH23 | BUSYCH22 | BUSYCH21 | BUSYCH20 | BUSYCH19 | BUSYCH18 | BUSYCH17 | BUSYCH16 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BUSYCH15 | BUSYCH14 | BUSYCH13 | BUSYCH12 | BUSYCH11 | BUSYCH10 | BUSYCH9 | BUSYCH8 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BUSYCH7 | BUSYCH6 | BUSYCH5 | BUSYCH4 | BUSYCH3 | BUSYCH2 | BUSYCH1 | BUSYCH0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – BUSYCHn** Busy Channel n [x=31..0]

This bit is cleared when the channel trigger action for DMA channel n is complete, when a bus error for DMA channel n is detected, or when DMA channel n is disabled.

This bit is set when DMA channel n starts a DMA transfer.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.12. Pending Channels

**Name:** PENDCH
**Offset:** 0x2C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PENDCH31 | PENDCH30 | PENDCH29 | PENDCH28 | PENDCH27 | PENDCH26 | PENDCH25 | PENDCH24 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PENDCH23 | PENDCH22 | PENDCH21 | PENDCH20 | PENDCH19 | PENDCH18 | PENDCH17 | PENDCH16 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PENDCH15 | PENDCH14 | PENDCH13 | PENDCH12 | PENDCH11 | PENDCH10 | PENDCH9 | PENDCH8 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PENDCH7 | PENDCH6 | PENDCH5 | PENDCH4 | PENDCH3 | PENDCH2 | PENDCH1 | PENDCH0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – PENDCH** Pending Channel n [n=31..0]

This bit is cleared when trigger execution defined by channel trigger action settings for DMA channel n is started, when a bus error for DMA channel n is detected or when DMA channel n is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.

This bit is set when a transfer is pending on DMA channel n.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.13. Active Channel and Levels

**Name:** ACTIVE
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BTCNT[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BTCNT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ABUSY | | | ID[4:0] | | | | |
| Access | R | | | R | R | R | R | R |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LVLEX3 | LVLEX2 | LVLEX1 | LVLEX0 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 31:16 – BTCNT[15:0]** Active Channel Block Transfer Count
These bits hold the 16-bit block transfer count of the ongoing transfer. This value is stored in the active channel and written back in the corresponding Write-Back channel memory location when the arbiter grants a new channel access. The value is valid only when the active channel Active Busy flag (ABUSY) is set.

**Bit 15 – ABUSY** Active Channel Busy
This bit is cleared when the active transfer count is written back in the write-back memory section.
This bit is set when the next descriptor transfer count is read from the write-back memory section.

**Bits 12:8 – ID[4:0]** Active Channel ID
These bits hold the channel index currently stored in the active channel registers. The value is updated each time the arbiter grants a new channel transfer access request.

**Bits 0, 1, 2, 3 – LVLEXx** Level x Channel Trigger Request Executing [x=3..0]
This bit is set when a level-x channel trigger request is executing or pending.
This bit is cleared when no request is pending or being executed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.14. Descriptor Memory Section Base Address

**Name:** BASEADDR
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** PAC Write Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BASEADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BASEADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BASEADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BASEADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – BASEADDR[31:0]** Descriptor Memory Base Address
These bits store the Descriptor memory section base address. The value must be 64-bit aligned.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.15. Write-Back Memory Section Base Address

**Name:** WRBADDR
**Offset:** 0x38
**Reset:** 0x00000000
**Property:** PAC Write Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – WRBADDR[31:0]** Write-Back Memory Base Address
These bits store the Write-Back memory base address. The value must be 64-bit aligned.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.16. Channel Control A

**Name:** CHCTRLA
**Offset:** 0x40 + n*0x04 [n=0..31]
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | THRESHOLD[1:0] | | BURSTLEN[3:0] | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | TRIGACT[1:0] | | | | | |
| Access | | | R/W | R/W | | | | |
| Reset | | | 0 | 0 | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | TRIGSRC[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | RUNSTDBY | | | | | ENABLE | SWRST |
| Access | | R/W | | | | | R/W | R/W |
| Reset | | 0 | | | | | 0 | 0 |

**Bits 29:28 – THRESHOLD[1:0]** FIFO Threshold
These bits define the threshold from which the DMA starts to write to the destination. These bits have no effect in the case of single beat transfers.
These bits are not enable-protected.

| Value | Name | Description |
|---|---|---|
| 0x0 | 1BEAT | Destination write starts after each beat source addess read |
| 0x1 | 2BEATS | Destination write starts after 2-beats source address read |
| 0x2 | 4BEATS | Destination write starts after 4-beats source address read |
| 0x3 | 8BEATS | Destination write starts after 8-beats source address read |

**Bits 27:24 – BURSTLEN[3:0]** Burst Length
These bits define the burst mode.
These bits are not enable-protected.

| Value | Name | Description |
|---|---|---|
| 0x0 | SINGLE | Single-beat burst |
| 0x1 | 2BEAT | 2-beats burst length |
| 0x2 | 3BEAT | 3-beats burst length |
| 0x3 | 4BEAT | 4-beats burst length |
| 0x4 | 5BEAT | 5-beats burst length |
| 0x5 | 6BEAT | 6-beats burst length |
| 0x6 | 7BEAT | 7-beats burst length |
| 0x7 | 8BEAT | 8-beats burst length |
| 0x8 | 9BEAT | 9-beats burst length |
| 0x9 | 10BEAT | 10-beats burst length |
| 0xA | 11BEAT | 11-beats burst length |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0xB | 12BEAT | 12-beats burst length |
| 0xC | 13BEAT | 13-beats burst length |
| 0xD | 14BEAT | 14-beats burst length |
| 0xE | 15BEAT | 15-beats burst length |
| 0xF | 16BEAT | 16-beats burst length |

**Bits 21:20 – TRIGACT[1:0]** Trigger Action

These bits define the trigger action used for a transfer.
These bits are not enable-protected.

| Value | Name | Description |
|---|---|---|
| 0x0 | BLOCK | One trigger required for each block transfer |
| 0x1 | - | Reserved |
| 0x2 | BURST | One trigger required for each burst transfer |
| 0x3 | TRANSACTION | One trigger required for each transaction |

**Bits 14:8 – TRIGSRC[6:0]** Trigger Source

These bits define the peripheral that will be the source of a trigger.

| Index | Instance | Channel | Presentation |
|---|---|---|---|
| 0x00 | DISABLE | - | Only software/event triggers |
| 0x01 | RTC | TIMESTAMP | DMA RTC timestamp trigger |
| 0x02 | DSU | DCC0 | DMAC ID for DCC0 register |
| 0x03 | DSU | DCC1 | DMAC ID for DCC1 register |
| 0x04 | SERCOM0 | RX | Index of DMA RX trigger |
| 0x05 | SERCOM0 | TX | Index of DMA TX trigger |
| 0x06 | SERCOM1 | RX | Index of DMA RX trigger |
| 0x07 | SERCOM1 | TX | Index of DMA TX trigger |
| 0x08 | SERCOM2 | RX | Index of DMA RX trigger |
| 0x09 | SERCOM2 | TX | Index of DMA TX trigger |
| 0x0A | SERCOM3 | RX | Index of DMA RX trigger |
| 0x0B | SERCOM3 | TX | Index of DMA TX trigger |
| 0x0C | SERCOM4 | RX | Index of DMA RX trigger |
| 0x0D | SERCOM4 | TX | Index of DMA TX trigger |
| 0x0E | SERCOM5 | RX | Index of DMA RX trigger |
| 0x0F | SERCOM5 | TX | Index of DMA TX trigger |
| 0x10 | SERCOM6 | RX | Index of DMA RX trigger |
| 0x11 | SERCOM6 | TX | Index of DMA TX trigger |
| 0x12 | SERCOM7 | RX | Index of DMA RX trigger |
| 0x13 | SERCOM7 | TX | Index of DMA TX trigger |
| 0x14 | CAN0 | DEBUG | DMA CAN Debug Req |
| 0x15 | CAN1 | DEBUG | DMA CAN Debug Req |
| 0x16 | TCC0 | OVF | DMA overflow/underflow/retrigger trigger |
| 0x1C - 0x17 | TCC0 | MC | Indexes of DMA Match/Compare triggers |
| 0x1D | TCC1 | OVF | DMA overflow/underflow/retrigger trigger |
| 0x21- 0x1E | TCC1 | MC | Indexes of DMA Match/Compare triggers |
| 0x22 | TCC2 | OVF | DMA overflow/underflow/retrigger trigger |
| 0x25 - 0x23 | TCC2 | MC | Indexes of DMA Match/Compare triggers |
| 0x26 | TCC3 | OVF | DMA overflow/underflow/retrigger trigger |
| 0x28 - 0x27 | TCC3 | MC | Indexes of DMA Match/Compare triggers |
| 0x29 | TCC4 | OVF | DMA overflow/underflow/retrigger trigger |
| 0x2B - 0x2A | TCC4 | MC | Indexes of DMA Match/Compare triggers |
| 0x2C | TC0 | OVF | Indexes of DMA Overflow trigger |
| 0x2E - 0x2D | TC0 | MC | Indexes of DMA Match/Compare triggers |
| 0x2F | TC1 | OVF | Indexes of DMA Overflow trigger |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| **TRIGSRC** (continued) | | | |
|---|---|---|---|
| **Index** | **Instance** | **Channel** | **Presentation** |
| 0x31 - 0x30 | TC1 | MC | Indexes of DMA Match/Compare triggers |
| 0x32 | TC2 | OVF | Indexes of DMA Overflow trigger |
| 0x34 - 0x33 | TC2 | MC | Indexes of DMA Match/Compare triggers |
| 0x35 | TC3 | OVF | Indexes of DMA Overflow trigger |
| 0x37 - 0x36 | TC3 | MC | Indexes of DMA Match/Compare triggers |
| 0x38 | TC4 | OVF | Indexes of DMA Overflow trigger |
| 0x3A - 0x39 | TC4 | MC | Indexes of DMA Match/Compare triggers |
| 0x3B | TC5 | OVF | Indexes of DMA Overflow trigger |
| 0x3D:0x3C | TC5 | MC | Indexes of DMA Match/Compare triggers |
| 0x3E | TC6 | OVF | Indexes of DMA Overflow trigger |
| 0x40 - 0x3F | TC6 | MC | Indexes of DMA Match/Compare triggers |
| 0x41 | TC7 | OVF | Indexes of DMA Overflow trigger |
| 0x43 - 0x41 | TC7 | MC | Indexes of DMA Match/Compare triggers |
| 0x44 | ADC0 | RESRDY | index of DMA RESRDY trigger |
| 0x45 | ADC0 | SEQ | Index of DMA SEQ trigger |
| 0x46 | ADC1 | RESRDY | Index of DMA RESRDY trigger |
| 0x47 | ADC1 | SEQ | Index of DMA SEQ trigger |
| 0x49 - 0x48 | DAC | EMPTY | DMA DAC Empty Req |
| 0x4B - 0x4A | DAC | RESRDY | DMA DAC Result Ready Req |
| 0x4D - 0x4C | I2S | RX | Indexes of DMA RX triggers |
| 0x4F - 0x4E | I2S | TX | Indexes of DMA TX triggers |
| 0x50 | PCC | RX | Indexes of PCC RX trigger |
| 0x51 | AES | WR | DMA DATA Write trigger |
| 0x52 | AES | RD | DMA DATA Read trigger |
| 0x53 | QSPI | RX | Indexes of QSPI RX trigger |
| 0x54 | QSPI | TX | Indexes of QSPI TX trigger |

**Bit 6 – RUNSTDBY** Channel run in standby

This bit is used to keep the DMAC channel running in standby mode.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The DMAC channel is halted in standby. |
| 1 | The DMAC channel continues to run in standby. |

**Bit 1 – ENABLE** Channel Enable

Writing a '0' to this bit during an ongoing transfer, the bit will not be cleared until the internal data transfer buffer is empty and the DMA transfer is aborted. The internal data transfer buffer will be empty once the ongoing burst transfer is completed.
Writing a '1' to this bit will enable the DMA channel.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | DMA channel is disabled. |
| 1 | DMA channel is enabled. |

**Bit 0 – SWRST** Channel Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the channel registers to their initial state. The bit can be set when the channel is disabled (ENABLE=0). Writing a '1' to this bit will be ignored as long as ENABLE=1. This bit is automatically cleared when the reset is completed.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.17. Channel Control B

**Name:** CHCTRLB
**Offset:** 0x44 + n*0x10 [n=0..31]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CMD[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – CMD[1:0]** Software Command
These bits define the software commands. Refer to Channel Suspend and Channel Resume and Next Suspend Skip.
These bits are not enable-protected.

| CMD[1:0] | Name | Description |
|---|---|---|
| 0x0 | NOACT | No action |
| 0x1 | SUSPEND | Channel suspend operation |
| 0x2 | RESUME | Channel resume operation |
| 0x3 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.18. Channel Priority Level

**Name:** CHPRILVL
**Offset:** 0x45 + n*0x10 [n=0..31]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | PRILVL[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – PRILVL[1:0]** Channel Priority Level
These bits define the priority level used for the DMA channel. The available levels are shown below, where a high level has priority over a low level. These bits are not enable-protected.

| Value | Name | Description |
|---|---|---|
| 0x0 | LVL0 | Channel Priority Level 0 (Lowest Level) |
| 0x1 | LVL1 | Channel Priority Level 1 |
| 0x2 | LVL2 | Channel Priority Level 2 |
| 0x3 | LVL3 | Channel Priority Level 3 (Highest Level) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.19. Channel Event Control

**Name:** CHEVCTRL
**Offset:** 0x46 + n*0x10 [n=0..31]
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EVOE | EVIE | EVOMODE[1:0] | | | EVACT[2:0] | | |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bit 7 – EVOE** Channel Event Output Enable
This bit indicates if the Channel event generation is enabled. The event will be generated for every condition defined in the Channel Event Output Selection bits (CHEVCTRL.EVOMODE).

| Value | Description |
|---|---|
| 0 | Channel event generation is disabled. |
| 1 | Channel event generation is enabled. |

**Bit 6 – EVIE** Channel Event Input Enable

| Value | Description |
|---|---|
| 0 | Channel event action will not be executed on any incoming event. |
| 1 | Channel event action will be executed on any incoming event. |

**Bits 5:4 – EVOMODE[1:0]** Channel Event Output Mode
These bits define the channel event output selection. For details on event output generation, refer to Event Output Selection.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Block event output selection. Refer to BTCTRL.EVOSEL for available selections. |
| 0x1 | TRIGACT | Ongoing trigger action |
| 0x2-0x3 | | Reserved |

**Bits 2:0 – EVACT[2:0]** Channel Event Input Action
These bits define the event input action. The action is executed only if the corresponding EVIE bit in the CHEVCTRL register of the channel is set. For details on event actions, refer to Event Input Actions. These bits are available only for channels with event input support.

| Value | Name | Description |
|---|---|---|
| 0x0 | NOACT | No action |
| 0x1 | TRIG | Transfer and periodic transfer trigger |
| 0x2 | CTRIG | Conditional transfer trigger |
| 0x3 | CBLOCK | Conditional block transfer |
| 0x4 | SUSPEND | Channel suspend operation |
| 0x5 | RESUME | Channel resume operation |
| 0x6 | SSKIP | Skip next block suspend action |
| 0x7 | INCPRI | Increase priority |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.20. Channel Interrupt Enable Clear

**Name:** CHINTENCLR
**Offset:** 0x4C + n*0x10 [n=0..31]
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Set (CHINTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SUSP | TCMPL | TERR |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – SUSP** Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Suspend Interrupt Enable bit, which disables the Channel Suspend interrupt.

| Value | Description |
|---|---|
| 0 | The Channel Suspend interrupt is disabled. |
| 1 | The Channel Suspend interrupt is enabled. |

**Bit 1 – TCMPL** Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Transfer Complete Interrupt Enable bit, which disables the Channel Transfer Complete interrupt.

| Value | Description |
|---|---|
| 0 | The Channel Transfer Complete interrupt is disabled. When block action is set to none, the TCMPL flag will not be set when a block transfer is completed. |
| 1 | The Channel Transfer Complete interrupt is enabled. |

**Bit 0 – TERR** Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Transfer Error Interrupt Enable bit, which disables the Channel Transfer Error interrupt.

| Value | Description |
|---|---|
| 0 | The Channel Transfer Error interrupt is disabled. |
| 1 | The Channel Transfer Error interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.21. Channel Interrupt Enable Set

**Name:** CHINTENSET
**Offset:** 0x4D + n*0x10 [n=0..31]
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Channel Interrupt Enable Clear (CHINTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SUSP | TCMPL | TERR |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – SUSP** Channel Suspend Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Suspend Interrupt Enable bit, which enables the Channel Suspend interrupt.

| Value | Description |
|---|---|
| 0 | The Channel Suspend interrupt is disabled. |
| 1 | The Channel Suspend interrupt is enabled. |

**Bit 1 – TCMPL** Channel Transfer Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Transfer Complete Interrupt Enable bit, which enables the Channel Transfer Complete interrupt.

| Value | Description |
|---|---|
| 0 | The Channel Transfer Complete interrupt is disabled. |
| 1 | The Channel Transfer Complete interrupt is enabled. |

**Bit 0 – TERR** Channel Transfer Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Channel Transfer Error Interrupt Enable bit, which enables the Channel Transfer Error interrupt.

| Value | Description |
|---|---|
| 0 | The Channel Transfer Error interrupt is disabled. |
| 1 | The Channel Transfer Error interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.22. Channel Interrupt Flag Status and Clear

**Name:** CHINTFLAG
**Offset:** 0x4E + n*0x10 [n=0..31]
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SUSP | TCMPL | TERR |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – SUSP** Channel Suspend
This flag is cleared by writing a '1' to it.
This flag is set when a block transfer with suspend block action is completed, when a software suspend command is executed, when a suspend event is received or when an invalid descriptor is fetched by the DMA.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel Suspend interrupt flag for the corresponding channel.
For details on available software commands, refer to CHCTRLB.CMD.
For details on available event input actions, refer to CHCTRLB.EVACT.
For details on available block actions, refer to BTCTRL.BLOCKACT.

**Bit 1 – TCMPL** Channel Transfer Complete
This flag is cleared by writing a '1' to it.
This flag is set when a block transfer is completed and the corresponding interrupt block action is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transfer Complete interrupt flag for the corresponding channel.

**Bit 0 – TERR** Channel Transfer Error
This flag is cleared by writing a '1' to it.
This flag is set when a bus error is detected during a beat transfer or when the DMAC fetches an invalid descriptor.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transfer Error interrupt flag for the corresponding channel.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.8.23. Channel Status

**Name:** CHSTATUS
**Offset:** 0x4F + n*0x10 [n=0..31]
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CRCERR | FERR | BUSY | PEND |
| Access | | | | | R/W | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 3 – CRCERR** Channel CRC Error
This bit is set when the CRC monitor detects data corruption. This bit is cleared bu writing '1' to it, or by clearing the CRC Error bit in the INTPEND register (INTPEND.CRCERR).

**Bit 2 – FERR** Channel Fetch Error
This bit is cleared when a software resume command is executed.
This bit is set when an invalid descriptor is fetched.

**Bit 1 – BUSY** Channel Busy
This bit is cleared when the channel trigger action is completed, when a bus error is detected or when the channel is disabled.
This bit is set when the DMA channel starts a DMA transfer.

**Bit 0 – PEND** Channel Pending
This bit is cleared when the channel trigger action is started, when a bus error is detected or when the channel is disabled. For details on trigger action settings, refer to CHCTRLB.TRIGACT.
This bit is set when a transfer is pending on the DMA channel, as soon as the transfer request is received.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 22.9. Register Summary - SRAM

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | BTCTRL | 7:0 | | | | BLOCKACT[1:0] | | EVOSEL[1:0] | | VALID |
| | | 15:8 | STEPSIZE[2:0] | | | STEPSEL | DSTINC | SRCINC | BEATSIZE[1:0] | |
| 0x02 | BTCNT | 7:0 | BTCNT[7:0] | | | | | | | |
| | | 15:8 | BTCNT[15:8] | | | | | | | |
| 0x04 | SRCADDR | 7:0 | SRCADDR[7:0] | | | | | | | |
| | | 15:8 | SRCADDR[15:8] | | | | | | | |
| | | 23:16 | SRCADDR[23:16] | | | | | | | |
| | | 31:24 | SRCADDR[31:24] | | | | | | | |
| 0x08 | DSTADDR | 7:0 | DSTADDR[7:0] | | | | | | | |
| | | 15:8 | DSTADDR[15:8] | | | | | | | |
| | | 23:16 | DSTADDR[23:16] | | | | | | | |
| | | 31:24 | DSTADDR[31:24] | | | | | | | |
| 0x0C | DESCADDR | 7:0 | DESCADDR[7:0] | | | | | | | |
| | | 15:8 | DESCADDR[15:8] | | | | | | | |
| | | 23:16 | DESCADDR[23:16] | | | | | | | |
| | | 31:24 | DESCADDR[31:24] | | | | | | | |

## 22.10. Register Description - SRAM

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.10.1. Block Transfer Control

**Name:** BTCTRL
**Offset:** 0x00
**Property:** -

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | STEPSIZE[2:0] | | | STEPSEL | DSTINC | SRCINC | BEATSIZE[1:0] | |
| Access | - | - | - | - | - | - | - | - |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | BLOCKACT[1:0] | | EVOSEL[1:0] | | VALID |
| Access | | | | - | - | - | - | - |
| Reset | | | | | | | | |

**Bits 15:13 – STEPSIZE[2:0]** Address Increment Step Size
These bits select the address increment step size. The setting apply to source or destination address, depending on STEPSEL setting.

| Value | Name | Description |
|---|---|---|
| 0x0 | X1 | Next ADDR = ADDR + (Beat size in byte) * 1 |
| 0x1 | X2 | Next ADDR = ADDR + (Beat size in byte) * 2 |
| 0x2 | X4 | Next ADDR = ADDR + (Beat size in byte) * 4 |
| 0x3 | X8 | Next ADDR = ADDR + (Beat size in byte) * 8 |
| 0x4 | X16 | Next ADDR = ADDR + (Beat size in byte) * 16 |
| 0x5 | X32 | Next ADDR = ADDR + (Beat size in byte) * 32 |
| 0x6 | X64 | Next ADDR = ADDR + (Beat size in byte) * 64 |
| 0x7 | X128 | Next ADDR = ADDR + (Beat size in byte) * 128 |

**Bit 12 – STEPSEL** Step Selection
This bit selects if source or destination addresses are using the step size settings.

| Value | Name | Description |
|---|---|---|
| 0x0 | DST | Step size settings apply to the destination address |
| 0x1 | SRC | Step size settings apply to the source address |

**Bit 11 – DSTINC** Destination Address Increment Enable
Writing a '0' to this bit will disable the destination address incrementation. The address will be kept fixed during the data transfer.
Writing a '1' to this bit will enable the destination address incrementation. By default, the destination address is incremented by 1. If the STEPSEL bit is cleared, flexible step-size settings are available in the STEPSIZE register.

| Value | Description |
|---|---|
| 0 | The Destination Address Increment is disabled |
| 1 | The Destination Address Increment is enabled |

**Bit 10 – SRCINC** Source Address Increment Enable
Writing a '0' to this bit will disable the source address incrementation. The address will be kept fixed during the data transfer.
Writing a '1' to this bit will enable the source address incrementation. By default, the source address is incremented by 1. If the STEPSEL bit is set, flexible step-size settings are available in the STEPSIZE register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The Source Address Increment is disabled |
| 1 | The Source Address Increment is enabled |

**Bits 9:8 – BEATSIZE[1:0]** Beat Size

These bits define the size of one beat. A beat is the size of one data transfer bus access, and the setting apply to both read and write accesses.

| Value | Name | Description |
|---|---|---|
| 0x0 | BYTE | 8-bit bus transfer |
| 0x1 | HWORD | 16-bit bus transfer |
| 0x2 | WORD | 32-bit bus transfer |
| other | | Reserved |

**Bits 4:3 – BLOCKACT[1:0]** Block Action

These bits define what actions the DMAC should take after a block transfer has completed.

| BLOCKACT[1:0] | Name | Description |
|---|---|---|
| 0x0 | NOACT | Channel will be disabled if it is the last block transfer in the transaction |
| 0x1 | INT | Channel will be disabled if it is the last block transfer in the transaction and block interrupt |
| 0x2 | SUSPEND | Channel suspend operation is completed |
| 0x3 | BOTH | Both channel suspend operation and block interrupt |

**Bits 2:1 – EVOSEL[1:0]** Event Output Selection

These bits define the event output selection.

| EVOSEL[1:0] | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Event generation disabled |
| 0x1 | BLOCK | Event strobe when block transfer complete |
| 0x2 | | Reserved |
| 0x3 | BEAT | Event strobe when beat transfer complete |

**Bit 0 – VALID** Descriptor Valid

Writing a '0' to this bit in the Descriptor or Write-Back memory will suspend the DMA channel operation when fetching the corresponding descriptor.
The bit is automatically cleared in the Write-Back memory section when channel is aborted, when an error is detected during the block transfer, or when the block transfer is completed.

| Value | Description |
|---|---|
| 0 | The descriptor is not valid |
| 1 | The descriptor is valid |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.10.2. Block Transfer Count

**Name:** BTCNT
**Offset:** 0x02
**Property:** -

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BTCNT[15:8] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BTCNT[7:0] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | | | | | | | | |

**Bits 15:0 – BTCNT[15:0]** Block Transfer Count
This bit group holds the 16-bit block transfer count.
During a transfer, the internal counter value is decremented by one after each beat transfer. The internal counter is written to the corresponding write-back memory section for the DMA channel when the DMA channel loses priority, is suspended or gets disabled. The DMA channel can be disabled by a complete transfer, a transfer error or by software.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.10.3. Block Transfer Source Address

**Name:** SRCADDR
**Offset:** 0x04
**Property:** -

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[31:24] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[23:16] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[15:8] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[7:0] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – SRCADDR[31:0]** Transfer Source Address

This bit field holds the block transfer source address.
When source address incrementation is disabled (BTCTRL.SRCINC=0), SRCADDR corresponds to the last beat transfer address in the block transfer.
When source address incrementation is enabled (BTCTRL.SRCINC=1), SRCADDR is calculated as follows:
If BTCTRL.STEPSEL=1:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

If BTCTRL.STEPSEL=0:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{START}$ is the source address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.10.4. Block Transfer Destination Address

**Name:** DSTADDR
**Offset:** 0x08
**Property:** -

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DSTADDR[31:24] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DSTADDR[23:16] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DSTADDR[15:8] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DSTADDR[7:0] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DSTADDR[31:0]** Transfer Destination Address

This bit field holds the block transfer destination address.
When destination address incrementation is disabled (BTCTRL.DSTINC = 0), DSTADDR corresponds to the last beat transfer address in the block transfer.
When destination address incrementation is enabled (BTCTRL.DSTINC = 1), DSTADDR is calculated as follows:
If BTCTRL.STEPSEL = 1:

$$DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$$

If BTCTRL.STEPSEL = 0:

$$DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$$

- $DSTADDR_{START}$ is the destination address of the first beat transfer in the block transfer
- BTCNT is the initial number of beats remaining in the block transfer
- BEATSIZE is the configured number of bytes in a beat
- STEPSIZE is the configured number of beats for each incrementation


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 22.10.5. Next Descriptor Address

**Name:** DESCADDR
**Offset:** 0x0C
**Property:** -

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DESCADDR[31:24] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DESCADDR[23:16] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DESCADDR[15:8] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DESCADDR[7:0] | | | | | | | |
| Access | - | - | - | - | - | - | - | - |
| Reset | | | | | | | | |

**Bits 31:0 – DESCADDR[31:0]** Next Descriptor Address
This bit group holds the SRAM address of the next descriptor. The value must be 128-bit aligned. If the value of this SRAM register is 0x00000000, the transaction will be terminated when the DMAC tries to load the next transfer descriptor.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 23. EIC – External Interrupt Controller

## 23.1. Overview

The External Interrupt Controller (EIC) allows external pins to be configured as interrupt lines. Each interrupt line can be individually masked and can generate an interrupt on rising, falling, both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Also, each external pin can be configured to be asynchronous in order to wake-up the device from Sleep modes where all clocks have been disabled. External pins can generate an event.

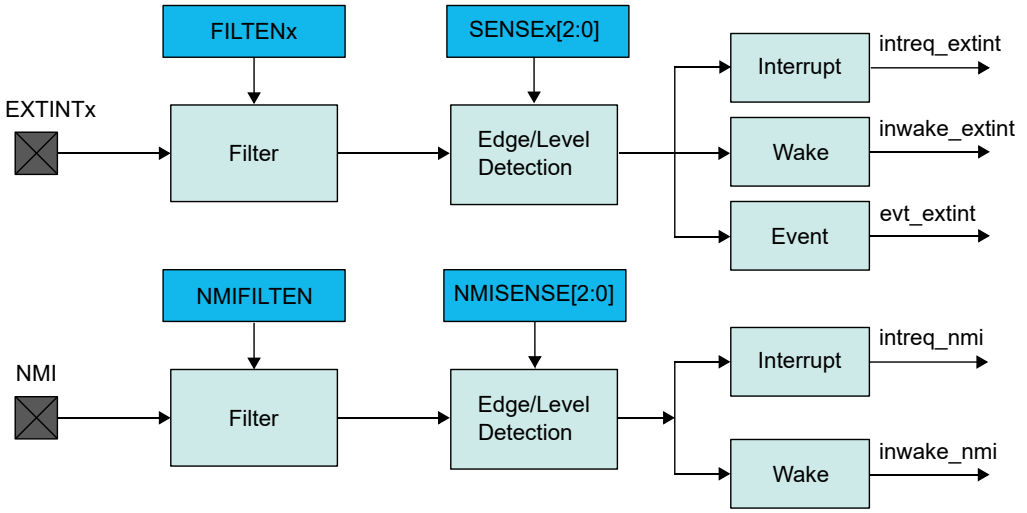
A separate Non-Maskable Interrupt (NMI) is supported. It has properties similar to the other external interrupts, but is connected to the NMI request of the CPU, enabling it to interrupt any other Interrupt mode.

## 23.2. Features

- Up to 16 external pins (EXTINTx), plus one non-maskable pin (NMI)
- Dedicated, Individually Maskable Interrupt for Each Pin
- Interrupt on Rising, Falling, or Both Edges
- Synchronous or Asynchronous Edge Detection mode
- Interrupt pin Debouncing
- Interrupt on High or Low Levels
- Asynchronous Interrupts for Sleep Modes Without Clock
- Filtering of External Pins
- Event Generation from EXTINTx

## 23.3. Block Diagram

**Figure 23-1.** EIC Block Diagram




## 23.4. Signal Description

| Signal Name | Type | Description |
|---|---|---|
| EXTINT[15..0] | Digital Input | External interrupt pin |
| NMI | Digital Input | Non-maskable interrupt pin |

One signal may be available on several pins.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 23.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 23.5.1. I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

**Related Links**
PORT - I/O Pin Controller

### 23.5.2. Power Management

All interrupts are available down to STANDBY Sleep mode, but the EIC can be configured to automatically mask some interrupts in order to prevent device wake-up.

The EIC will continue to operate in any Sleep mode where the selected source clock is running. The EIC's interrupts can be used to wake up the device from Sleep modes. Events connected to the Event System can trigger other operations in the system without exiting Sleep modes.

**Related Links**
PM – Power Manager

### 23.5.3. Clocks

The EIC bus clock (CLK_EIC_APB) can be enabled and disabled by the Main Clock Controller, the default state of CLK_EIC_APB can be found in the Peripheral Clock Masking section.

Some optional functions need a peripheral clock, which can either be a generic clock (GCLK_EIC, for wider frequency selection) or a Ultra Low-Power 32 KHz clock (CLK_ULP32K, for highest power efficiency). One of the clock sources must be configured and enabled before using the peripheral:

GCLK_EIC is configured and enabled in the Generic Clock Controller.

CLK_ULP32K is provided by the internal Ultra Low-Power (OSCULP32K) Oscillator in the OSC32KCTRL module.

Both GCLK_EIC and CLK_ULP32K are asynchronous to the user interface clock (CLK_EIC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**
MCLK – Main Clock
Peripheral Clock Masking
GCLK - Generic Clock Controller
OSC32KCTRL – 32KHz Oscillators Controller

### 23.5.4. DMA

Not applicable.

### 23.5.5. Interrupts

There are several interrupt request lines, some (the number depends on the product variant) for the external interrupts (EXTINT) and one for Non-Maskable Interrupt (NMI).

Each EXTINT interrupt request line is connected to the interrupt controller. Using the EIC interrupt requires the interrupt controller to be configured first.

The NMI interrupt request line is connected to the interrupt controller, but does not require the interrupt to be configured.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Nested Vector Interrupt Controller

### 23.5.6. Events

The events are connected to the Event System. Using the events requires the Event System to be configured first.

**Related Links**
Event System (EVSYS)

### 23.5.7. Debug Operation

When the CPU is halted in Debug mode, the EIC continues normal operation. If the EIC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

### 23.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)
- Non-Maskable Interrupt Flag Status and Clear register (NMIFLAG)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 23.5.9. Analog Connections

Not applicable.

## 23.6. Functional Description

### 23.6.1. Principle of Operation

The EIC detects edge or level condition to generate interrupts to the CPU interrupt controller or events to the Event System. Each external interrupt pin (EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K.

**Related Links**
External Pin Processing

### 23.6.2. Basic Operation

#### 23.6.2.1. Initialization

The EIC must be initialized in the following order:

1. Enable CLK_EIC_APB.
2. If required, configure the NMI by writing the Non-Maskable Interrupt Control register (NMICTRL).
3. Enable GCLK_EIC or CLK_ULP32K when one of the following configuration is selected:
   - The NMI uses edge detection or filtering
   - One or more EXTINT uses filtering
   - One or more EXTINT uses synchronous edge detection


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- One or more EXTINT uses debouncing

GCLK_EIC is used when a frequency higher than 32.768 kHz is required for filtering.

CLK_ULP32K is recommended when power consumption is the priority. For CLK_ULP32K write a '1' to the Clock Selection bit in the Control A register (CTRLA.CKSEL).

4. Configure the EIC input sense and filtering by writing the configuration register (CONFIG0 or CONFIG1).
5. Optionally, enable the asynchronous mode.
6. Optionally, enable the debouncer mode.
7. Enable the EIC by writing a '1' to CTRLA.ENABLE.

The following bits are enable-protected, that is, it can only be written when the EIC is disabled (CTRLA.ENABLE = 0):

- Clock Selection bit in the Control A register (CTRLA.CKSEL)

The following registers are enable-protected:

- Event Control register (EVCTRL)
- Configuration register (CONFIGn)
- External Interrupt Asynchronous Mode register (ASYNCH)
- Debouncer Enable register (DEBOUNCEN)
- Debounce Prescaler register (DPRESCALER)

Enable-protected bits in the CTRLA register can be written simultaneously when setting CTRLA.ENABLE to '1', but not at the same time as CTRLA.ENABLE is being cleared.

Enable-protection is denoted by the "Enable-Protected" property in the register description.

#### 23.6.2.2. Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing CTRLA.ENABLE to '0'.

The EIC is reset by setting the Software Reset bit in the Control register (CTRLA.SWRST). All registers in the EIC will be reset to their initial state, and the EIC will be disabled.

Refer to the CTRLA register description for details.

### 23.6.3. External Pin Processing

Each external pin can be configured to generate an interrupt/event on edge detection (rising, falling or both edges) or level detection (high or low). The sense of external interrupt pins is configured by writing the Input Sense x bits in the Configuration register (CONFIGn.SENSEx). The corresponding interrupt flag (INTFLAG.EXTINT[x]) in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition is met.

When the interrupt flag has been cleared in edge-sensitive mode, INTFLAG.EXTINT[x] will only be set if a new interrupt condition is met.

In level-sensitive mode, when interrupt has been cleared, INTFLAG.EXTINT[x] will be set immediately if the EXTINTx pin still matches the interrupt condition.

Each external pin can be filtered by a majority vote filtering, clocked by GCLK_EIC or CLK_ULP32K. Filtering is enabled if bit Filter Enable x in the configuration register (CONFIGn.FILTENx) is written to '1'. The majority vote filter samples the external pin three times with GCLK_EIC or CLK_ULP32K and outputs the value when two or more samples are equal.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

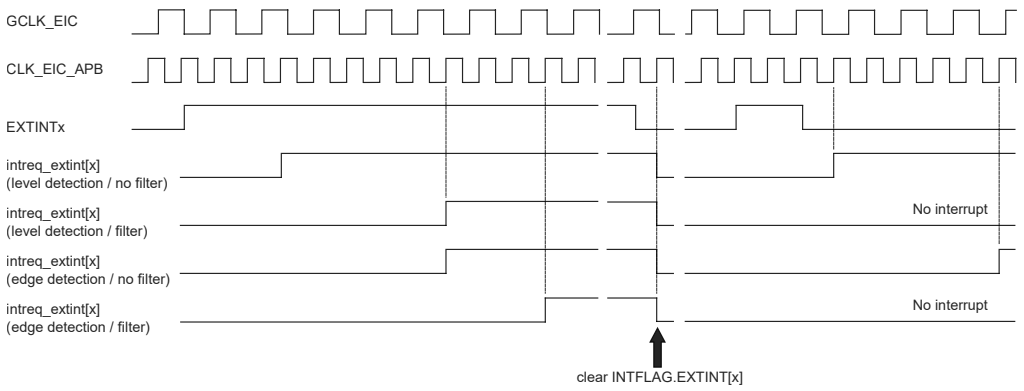
**Table 23-1.** Majority Vote Filter Logic

| Samples [0, 1, 2] | Filter Output |
|---|---|
| [0,0,0] | 0 |
| [0,0,1] | 0 |
| [0,1,0] | 0 |
| [0,1,1] | 1 |
| [1,0,0] | 0 |
| [1,0,1] | 1 |
| [1,1,0] | 1 |
| [1,1,1] | 1 |

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Level detection and asynchronous edge detection do not require GCLK_EIC or CLK_ULP32K, and can generate asynchronous interrupts and events.

If filtering, synchronous edge detection or debouncing is enabled, the EIC automatically requests GCLK_EIC or CLK_ULP32K to operate. The selection between these two clocks is done by writing the Clock Selection bits in the Control A register (CTRLA.CKSEL). GCLK_EIC must be enabled in the GCLK module. In these modes the external pin is sampled at the EIC clock rate, thus pulses with duration lower than two EIC clock periods may not be properly detected.

**Figure 23-2.** Interrupt Detection Latency by modes (Rising Edge)




Detection latency depends on the detection mode.

**Table 23-2.** Detection Latency

| Detection mode | Latency (worst case) |
|---|---|
| Level without filter | Five CLK_EIC_APB periods |
| Level with filter | Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods |
| Edge without filter | Four GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods |
| Edge with filter | Six GCLK_EIC/CLK_ULP32K periods + five CLK_EIC_APB periods |

### Related Links

GCLK - Generic Clock Controller

## 23.6.4. Additional Features

### 23.6.4.1. Non-Maskable Interrupt (NMI)

The non-maskable interrupt pin can also generate an interrupt on edge or level detection, but it is configured with the dedicated NMI Control register (NMICTRL). To select the sense for NMI, write to


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

the NMISENSE bit group in the NMI Control register (NMICTRL.NMISENSE). NMI filtering is enabled by writing a '1' to the NMI Filter Enable bit (NMICTRL.NMIFILTEN).

If edge detection or filtering is required, enable GCLK_EIC or CLK_ULP32K.

NMI detection is enabled only by the NMICTRL.NMISENSE value, and the EIC is not required to be enabled.

When an NMI is detected, the Non-maskable Interrupt flag in the NMI Flag Status and Clear register is set (NMIFLAG.NMI). NMI interrupt generation is always enabled, and NMIFLAG.NMI generates an interrupt request when set.

#### 23.6.4.2. Asynchronous Edge Detection Mode (No Debouncing)

The EXTINT edge detection can be operated synchronously or asynchronously, selected by the Asynchronous Control Mode bit for external pin x in the External Interrupt Asynchronous Mode register (ASYNCH.ASYNCH[x]). The EIC edge detection is operated synchronously when the Asynchronous Control Mode bit (ASYNCH.ASYNCH[x]) is '0' (default value). It is operated asynchronously when ASYNCH.ASYNCH[x] is written to '1'.

In *Synchronous Edge Detection Mode*, the external interrupt (EXTINT) or the non-maskable interrupt (NMI) pins are sampled using the EIC clock as defined by the Clock Selection bit in the Control A register (CTRLA.CKSEL). The External Interrupt flag (INTFLAG.EXTINT[x]) or Non-Maskable Interrupt flag (NMIFLAG.NMI) is set when the last sampled state of the pin differs from the previously sampled state. In this mode, the EIC clock is required.

The Synchronous Edge Detection Mode can be used in Idle and Standby sleep modes.

In *Asynchronous Edge Detection Mode*, the external interrupt (EXTINT) pins or the non-maskable interrupt (NMI) pins set the External Interrupt flag or Non-Maskable Interrupt flag (INTFLAG.EXTINT[x] or NMIFLAG) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

#### 23.6.4.3. Interrupt Pin Debouncing

The external interrupt pin (EXTINT) edge detection can use a debouncer to improve input noise immunity. When selected, the debouncer can work in the synchronous mode or the asynchronous mode, depending on the configuration of the ASYNCH.ASYNCH[x] bit for the pin. The debouncer uses the EIC clock as defined by the bit CTRLA.CKSEL to clock the debouncing circuitry. The debouncing time frame is set with the debouncer prescaler DPRESCALER.DPRESCALERn, which provides the *low frequency clock* tick that is used to reject higher frequency signals.

The debouncing mode for pin EXTINT x can be selected only if the Sense bits in the Configuration y register (CONFIGy.SENSEx) are set to RISE, FALL or BOTH. If the debouncing mode for pin EXTINT x is selected, the filter mode for that pin (CONFIGy.FILTENx) can not be selected.

The debouncer manages an internal “valid pin state” that depends on the external interrupt (EXTINT) pin transitions, the debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin (EXTINT) is sampled continously on EIC clock. The sampled value is evaluated on each *low frequency clock* tick to detect a transitional edge when the sampled value is different of the current valid pin state. The sampled value is evaluated on each EIC clock when DPRESCALER.TICKON=0 or on each *low frequency clock* tick when DPRESCALER.TICKON=1, to detect a bounce when the sampled value is equal to the current valid pin state. Transitional edge detection increments the transition counter of the EXTINT pin, while bounce detection resets the transition counter. The transition counter must exceed the transition count threshold as defined by the DPRESCALER.STATESn bitfield. In the synchronous mode the threshold is 4 when DPRESCALER.STATESn=0 or 8 when DPRESCALER.STATESn=1. In the asynchronous mode the threshold is 4.

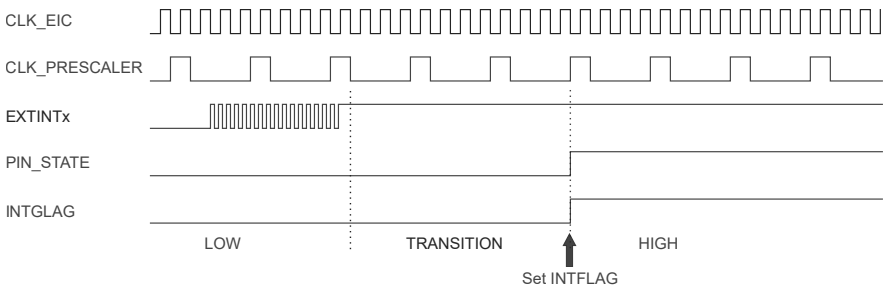
The valid pin state for the pins can be accessed by reading the register PINSTATE for both synchronous or asynchronous debouncing mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Synchronous edge detection** In this mode the external interrupt (EXTINT) pin is sampled continously on EIC clock.

1. A pin edge transition will be validated when the sampled value is consistently different of the current valid pin state for 4 (or 8 depending on bit DPRESCALER.STATESn) consecutive ticks of the low frequency clock.
2. Any pin sample, at the *low frequency clock* tick rate, with a value opposite to the current valid pin state will increment the transition counter.
3. Any pin sample, at EIC clock rate (when DPRESCALER.TICKON=0) or the *low frequency clock* tick (when DPRESCALER.TICKON=1), with a value identical to the current valid pin state will return the transition counter to zero.
4. When the transition counter meets the count threshold, the pin edge transition is validated and the pin state PINSTATE.PINSTATE[x] is changed to the detected level.
5. The external interrupt flag (INTFLAG.EXTINT[x]) is set when the pin state PINSTATE.PINSTATE[x] is changed.

**Figure 23-3.** EXTINT Pin Synchronous Debouncing (Rising Edge)

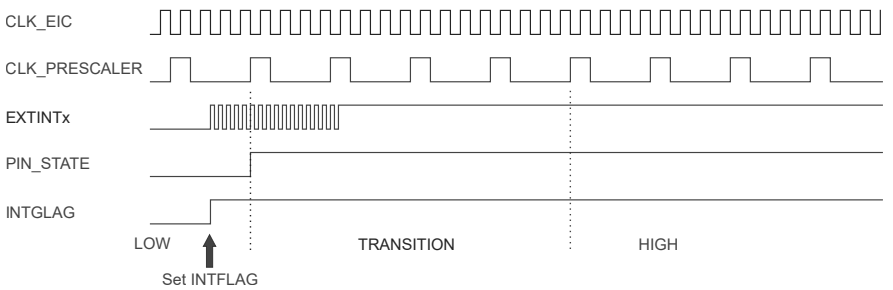



In the synchronous edge detection mode, the EIC clock is required. The synchronous edge detection mode can be used in Idle and Standby sleep modes.

**Asynchronous edge detection** In this mode, the external interrupt (EXTINT) pin directly drives an asynchronous edges detector which triggers any rising or falling edge on the pin:

1. Any edge detected that indicates a transition from the current valid pin state will immediately set the valid pin state PINSTATE.PINSTATE[x] to the detected level.
2. The external interrupt flag (INTFLAG.EXTINT[x] is immediately changed.
3. The edge detector will then be idle until no other rising or falling edge transition is detected during 4 consecutive ticks of the low frequency clock.
4. Any rising or falling edge transition detected during the idle state will return the transition counter to 0.
5. After 4 consecutive ticks of the low frequency clock without bounce detected, the edge detector is ready for a new detection.

**Figure 23-4.** EXTINT Pin Asynchronous Debouncing (Rising Edge)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

In this mode, the EIC clock is requested. The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

### 23.6.5. DMA Operation

Not applicable.

### 23.6.6. Interrupts

The EIC has the following interrupt sources:

- External interrupt pins (EXTINTx). See Basic Operation.
- Non-maskable interrupt pin (NMI). See Additional Features.

Each interrupt source has an associated Interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when an Interrupt condition occurs (NMIFLAG for NMI). Each interrupt, except NMI, can be individually enabled by setting the corresponding bit in the Interrupt Enable Set register (INTENSET=1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR=1).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the EIC is reset. See the INTFLAG register for details on how to clear Interrupt flags. The EIC has one interrupt request line for each external interrupt (EXTINTx) and one line for NMI. The user must read the INTFLAG (or NMIFLAG) register to determine which Interrupt condition is present.

**Notes:**

1. Interrupts must be globally enabled for interrupt requests to be generated.
2. If an external interrupts (EXTINT) is common on two or more I/O pins, only one will be active (the first one programmed).

**Related Links**

Processor and Architecture

### 23.6.7. Events

The EIC can generate the following output events:

- External event from pin (EXTINTx).

Setting an Event Output Control register (EVCTRL.EXTINTEO) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to *Event System* for details on configuring the Event System.

When the condition on pin EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

**Related Links**

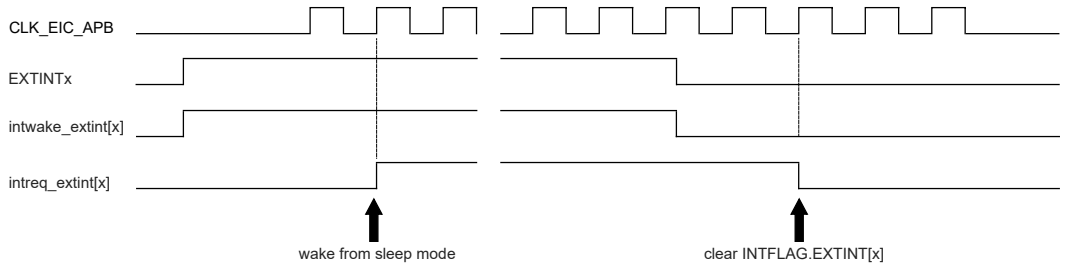
Event System (EVSYS)

### 23.6.8. Sleep Mode Operation

In sleep modes, an EXTINTx pin can wake up the device if the corresponding condition matches the configuration in the CONFIG register, and the corresponding bit in the Interrupt Enable Set register (INTENSET) is written to '1'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 23-5.** Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)




### 23.6.9. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 23.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | CKSEL | | | ENABLE | SWRST |
| 0x01 | NMICTRL | 7:0 | | | | NMIASYNCH | NMIFILTEN | NMISENSE[2:0] | | |
| 0x02 | NMIFLAG | 7:0 | | | | | | | | NMI |
| | | 15:8 | | | | | | | | |
| 0x04 | SYNCBUSY | 7:0 | | | | | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | EVCTRL | 7:0 | EXTINTEO[7:0] | | | | | | | |
| | | 15:8 | EXTINTEO[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | INTENCLR | 7:0 | EXTINT[7:0] | | | | | | | |
| | | 15:8 | EXTINT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | INTENSET | 7:0 | EXTINT[7:0] | | | | | | | |
| | | 15:8 | EXTINT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | INTFLAG | 7:0 | EXTINT[7:0] | | | | | | | |
| | | 15:8 | EXTINT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | ASYNCH | 7:0 | ASYNCH[7:0] | | | | | | | |
| | | 15:8 | ASYNCH[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | CONFIG0 | 7:0 | FILTEN1 | SENSE1[2:0] | | | FILTEN0 | SENSE0[2:0] | | |
| | | 15:8 | FILTEN3 | SENSE3[2:0] | | | FILTEN2 | SENSE2[2:0] | | |
| | | 23:16 | FILTEN5 | SENSE5[2:0] | | | FILTEN4 | SENSE4[2:0] | | |
| | | 31:24 | FILTEN7 | SENSE7[2:0] | | | FILTEN6 | SENSE6[2:0] | | |
| 0x20 | CONFIG1 | 7:0 | FILTEN9 | SENSE9[2:0] | | | FILTEN8 | SENSE8[2:0] | | |
| | | 15:8 | FILTEN11 | SENSE11[2:0] | | | FILTEN10 | SENSE10[2:0] | | |
| | | 23:16 | FILTEN13 | SENSE13[2:0] | | | FILTEN12 | SENSE12[2:0] | | |
| | | 31:24 | FILTEN15 | SENSE15[2:0] | | | FILTEN14 | SENSE14[2:0] | | |
| 0x24 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DEBOUNCEN | 7:0 | DEBOUNCEN[7:0] | | | | | | | |
| | | 15:8 | DEBOUNCEN[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | DPRESCALER | 7:0 | STATES1 | PRESCALER1[2:0] | | | STATES0 | PRESCALER0[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | TICKON |
| | | 31:24 | | | | | | | | |
| 0x38 | PINSTATE | 7:0 | PINSTATE[7:0] | | | | | | | |
| | | 15:8 | PINSTATE[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 23.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 23.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | CKSEL | | | ENABLE | SWRST |
| Access | | | | RW | | | RW | W |
| Reset | | | | 0 | | | 0 | 0 |

**Bit 4 – CKSEL** Clock Selection
The EIC can be clocked either by GCLK_EIC (when a frequency higher than 32.768 KHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).
This bit is not Write-Synchronized.

| Value | Description |
|---|---|
| 0 | The EIC is clocked by GCLK_EIC. |
| 1 | The EIC is clocked by CLK_ULP32K. |

**Bit 1 – ENABLE** Enable
Due to synchronization there is a delay between writing to CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable bit in the Synchronization Busy register will be set (SYNCBUSY.ENABLE=1). SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

| Value | Description |
|---|---|
| 0 | The EIC is disabled. |
| 1 | The EIC is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the EIC to their initial state, and the EIC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not Enable-Protected.
This bit is Write-Synchronized.

| Value | Description |
|---|---|
| 0 | There is no ongoing reset operation. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.2. Non-Maskable Interrupt Control

**Name:** NMICTRL
**Offset:** 0x01
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | NMIASYNCH | NMIFILTEN | NMISENSE[2:0] | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 4 – NMIASYNCH** Non-Maskable Interrupt Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

| Value | Description |
|---|---|
| 0 | The NMI edge detection is synchronously operated. |
| 1 | The NMI edge detection is asynchronously operated. |

**Bit 3 – NMIFILTEN** Non-Maskable Interrupt Filter Enable

| Value | Description |
|---|---|
| 0 | NMI filter is disabled. |
| 1 | NMI filter is enabled. |

**Bits 2:0 – NMISENSE[2:0]** Non-Maskable Interrupt Sense Configuration
These bits define on which edge or level the NMI triggers.

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No detection |
| 0x1 | RISE | Rising-edge detection |
| 0x2 | FALL | Falling-edge detection |
| 0x3 | BOTH | Both-edge detection |
| 0x4 | HIGH | High-level detection |
| 0x5 | LOW | Low-level detection |
| 0x6 – 0x7 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.3. Non-Maskable Interrupt Flag Status and Clear

**Name:** NMIFLAG
**Offset:** 0x02
**Reset:** 0x0000

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | NMI |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – NMI** Non-Maskable Interrupt
This flag is cleared by writing a '1' to it.
This flag is set when the NMI pin matches the NMI sense configuration, and will generate an interrupt request.
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.4. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x04
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ENABLE** Enable Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization for CTRLA.ENABLE bit is ongoing. |

**Bit 0 – SWRST** Software Reset Synchronization Busy Status

| Value | Description |
|---|---|
| 0 | Write synchronization for CTRLA.SWRST bit is complete. |
| 1 | Write synchronization for CTRLA.SWRST bit is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.5. Event Control

**Name:** EVCTRL
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXTINTEO[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EXTINTEO[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – EXTINTEO[15:0]** External Interrupt Event Output Enable
The bit x of EXTINTEO enables the event associated with the EXTINTx pin.

| Value | Description |
|---|---|
| 0 | Event from pin EXTINTx is disabled. |
| 1 | Event from pin EXTINTx is enabled and will be generated when EXTINTx pin matches the external interrupt sensing configuration. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.6. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – EXTINT[15:0]** External Interrupt Enable
The bit x of EXTINT disables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will clear the External Interrupt Enable bit x, which disables the external interrupt EXTINTx.

| Value | Description |
|---|---|
| 0 | The external interrupt x is disabled. |
| 1 | The external interrupt x is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.7. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – EXTINT[15:0]** External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EXTINTx pin.
Writing a '0' to bit x has no effect.
Writing a '1' to bit x will set the External Interrupt Enable bit x, which enables the external interrupt EXTINTx.

| Value | Description |
|---|---|
| 0 | The external interrupt x is disabled. |
| 1 | The external interrupt x is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.8. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – EXTINT[15:0]** External Interrupt

The flag bit x is cleared by writing a '1' to it.
This flag is set when EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR/SET.EXTINT[x] is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the External Interrupt x flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.9. External Interrupt Asynchronous Mode

**Name:** ASYNCH
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ASYNCH[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ASYNCH[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – ASYNCH[15:0]** Asynchronous Edge Detection Mode
The bit x of ASYNCH set the Asynchronous Edge Detection Mode for the interrupt associated with the EXTINTx pin.

| Value | Description |
|---|---|
| 0 | The EXTINT x edge detection is synchronously operated. |
| 1 | The EXTINT x edge detection is asynchronously operated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.10. External Interrupt Sense Configuration n

**Name:** CONFIG0
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN7 | SENSE7[2:0] | | | FILTEN6 | SENSE6[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN5 | SENSE5[2:0] | | | FILTEN4 | SENSE4[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN3 | SENSE3[2:0] | | | FILTEN2 | SENSE2[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN1 | SENSE1[2:0] | | | FILTEN0 | SENSE0[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 3, 7, 11, 15, 19, 23, 27, 31 – FILTENx** Filter Enable x [x=7..0]
**Note:** The filter must be disabled if the asynchronous detection is enabled.

| Value | Description |
|---|---|
| 0 | Filter is disabled for EXTINT[x] input. |
| 1 | Filter is enabled for EXTINT[x] input. |

**Bits 0:2, 4:6, 8:10, 12:14, 16:18, 20:22, 24:26, 28:30 – SENSEx** Input Sense Configuration x [x=7..0]
These bits define on which edge or level the interrupt or event for EXTINT[x] will be generated.

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No detection |
| 0x1 | RISE | Rising-edge detection |
| 0x2 | FALL | Falling-edge detection |
| 0x3 | BOTH | Both-edge detection |
| 0x4 | HIGH | High-level detection |
| 0x5 | LOW | Low-level detection |
| 0x6 – 0x7 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.11. External Interrupt Sense Configuration n

**Name:** CONFIG1
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN15 | SENSE15[2:0] | | | FILTEN14 | SENSE14[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN13 | SENSE13[2:0] | | | FILTEN12 | SENSE12[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN11 | SENSE11[2:0] | | | FILTEN10 | SENSE10[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FILTEN9 | SENSE9[2:0] | | | FILTEN8 | SENSE8[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 3, 7, 11, 15, 19, 23, 27, 31 – FILTENx** Filter Enable x [x=15..8]
**Note:** The filter must be disabled if the asynchronous detection is enabled.

| Value | Description |
|---|---|
| 0 | Filter is disabled for EXTINT[x] input. |
| 1 | Filter is enabled for EXTINT[x] input. |

**Bits 0:2, 4:6, 8:10, 12:14, 16:18, 20:22, 24:26, 28:30 – SENSEx** Input Sense Configuration x [x=15..8]
These bits define on which edge or level the interrupt or event for EXTINT[x] will be generated.

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No detection |
| 0x1 | RISE | Rising-edge detection |
| 0x2 | FALL | Falling-edge detection |
| 0x3 | BOTH | Both-edge detection |
| 0x4 | HIGH | High-level detection |
| 0x5 | LOW | Low-level detection |
| 0x6 – 0x7 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.12. Debouncer Enable

**Name:** DEBOUNCEN
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DEBOUNCEN[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DEBOUNCEN[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DEBOUNCEN[15:0]** Debouncer Enable
The bit x of DEBOUNCEN set the Debounce mode for the interrupt associated with the EXTINTx pin.

| Value | Description |
|---|---|
| 0 | The EXTINT x edge input is not debounced. |
| 1 | The EXTINT x edge input is debounced. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.13. Debouncer Prescaler

**Name:** DPRESCALER
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | TICKON |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | STATES1 | PRESCALER1[2:0] | | | STATES0 | PRESCALER0[2:0] | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 16 – TICKON** Pin Sampler frequency selection
This bit selects the clock used for the sampling of bounce during transition detection.

| Value | Description |
|---|---|
| 0 | The bounce sampler is using GCLK_EIC. |
| 1 | The bounce sampler is using the low frequency clock. |

**Bits 3, 7 – STATESx** Debouncer number of states x
This bit selects the number of samples by the debouncer low frequency clock needed to validate a transition from current pin state to next pin state in synchronous debouncing mode for pins EXTINT[7+(8x):8x].

| Value | Description |
|---|---|
| 0 | The number of low frequency samples is 3. |
| 1 | The number of low frequency samples is 7. |

**Bits 0:2, 4:6 – PRESCALERx** Debouncer Prescaler x
These bits select the debouncer low frequency clock for pins EXTINT[7+(8x):8x].

| Value | Name | Description |
|---|---|---|
| 0x0 | F/2 | EIC clock divided by 2 |
| 0x1 | F/4 | EIC clock divided by 4 |
| 0x2 | F/8 | EIC clock divided by 8 |
| 0x3 | F/16 | EIC clock divided by 16 |
| 0x4 | F/32 | EIC clock divided by 32 |
| 0x5 | F/64 | EIC clock divided by 64 |
| 0x6 | F/128 | EIC clock divided by 128 |
| 0x7 | F/256 | EIC clock divided by 256 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 23.8.14. Pin State

**Name:** PINSTATE
**Offset:** 0x38
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PINSTATE[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PINSTATE[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – PINSTATE[15:0]** Pin State
These bits return the valid pin state of the debounced external interrupt pin EXTINTx.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 24. GMAC - Ethernet MAC

The description and registers of this peripheral are using the 'GMAC' designation although the device does not support Gigabit Ethernet functionality.

## 24.1. Description

The Ethernet Media Access Controller (GMAC) module implements a 10/100 Mbps Ethernet MAC, compatible with the IEEE 802.3 standard. The GMAC can operate in either half or full duplex mode at all supported speeds.

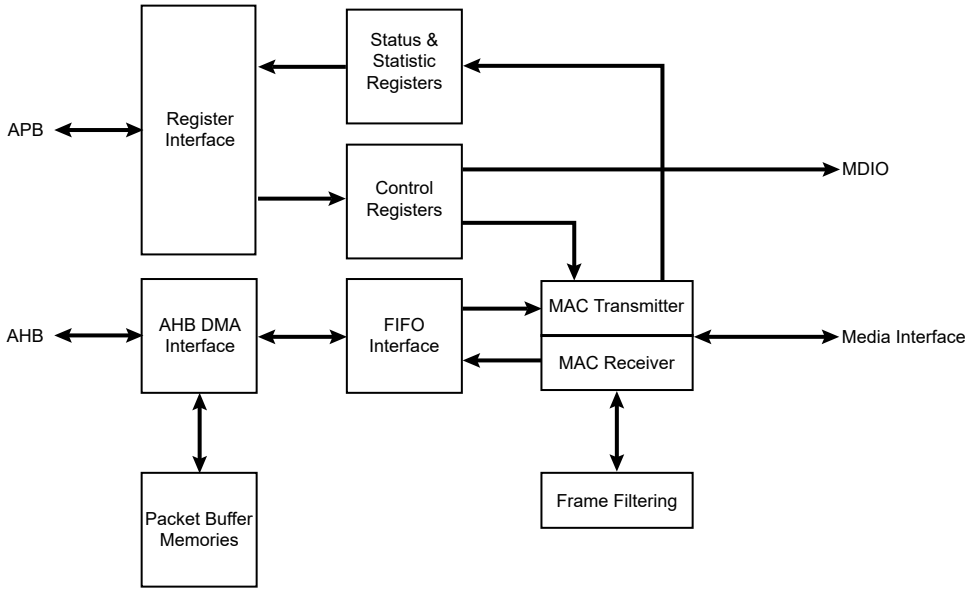
## 24.2. Features

- Compatible with IEEE Standard 802.3
- 10, 100 Mbps operation
- Full and half duplex operation at all supported speeds of operation
- Statistics Counter Registers for RMON/MIB
- MII/RMII interface to the physical layer
- Integrated physical coding
- Direct memory access (DMA) interface to external memory
- Programmable burst length and endianism for DMA
- Interrupt generation to signal receive and transmit completion, errors or other events
- Automatic pad and cyclic redundancy check (CRC) generation on transmitted frames
- Automatic discard of frames received with errors
- Receive and transmit IP, TCP, and UDP checksum offload. Both IPv4 and IPv6 packet types supported
- Address checking logic for four specific 48-bit addresses, four type IDs, promiscuous mode, hash matching of unicast and multicast destination addresses, and Wake-on-LAN
- Management Data Input/Output (MDIO) interface for physical layer management
- Support for jumbo frames up to 10240 Bytes
- Full duplex flow control with recognition of incoming pause frames and hardware generation of transmitted pause frames
- Half duplex flow control by forcing collisions on incoming frames
- Support for 802.1Q VLAN tagging with recognition of incoming VLAN and priority tagged frames
- Programmable Inter Packet Gap (IPG) Stretch
- Recognition of IEEE 1588 PTP frames
- IEEE 1588 time stamp unit (TSU) and TSU event generation
- Support for 802.1AS timing and synchronization
- Supports 802.1Qav traffic shaping on two highest priority queues
- Support for 802.3az Energy Efficient Ethernet


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 24.3. Block Diagram

**Figure 24-1.** Block Diagram




## 24.4. Signal Description

The GMAC includes the following signal interfaces:

- MII, RMII to an external PHY
- MDIO interface for external PHY management
- Client APB interface for accessing GMAC registers
- Host AHB interface for memory access
- Internal GTSUCOMP signal for TSU timer count value comparison

**Table 24-1.** GMAC Connections in Different Modes

| Signal Name | Function | MII | RMII |
|---|---|---|---|
| GTXCK | Transmit Clock or Reference Clock | TXCK | REFCK |
| GTXEN | Transmit Enable | TXEN | TXEN |
| GTX[3..0] | Transmit Data | TXD[3:0] | TXD[1:0] |
| GTXER | Transmit Coding Error | TXER | Not Used |
| GRXCK | Receive Clock | RXCK | Not Used |
| GRXDV | Receive Data Valid | RXDV | CRSDV |
| GRX[3..0] | Receive Data | RXD[3:0] | RXD[1:0] |
| GRXER | Receive Error | RXER | RXER |
| GCRS | Carrier Sense and Data Valid | CRS | Not Used |
| GCOL | Collision Detect | COL | Not Used |
| GMDC | Management Data Clock | MDC | MDC |
| GMDIO | Management Data Input/Output | MDIO | MDIO |

## 24.5. Product Dependencies

### 24.5.1. I/O Lines

Using the GMAC I/O lines requires the I/O pins to be configured using the port configuration (PORT).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
I/O Multiplexing and Considerations
PORT - I/O Pin Controller

### 24.5.2. Power Management

The GMAC continues to operate in IDLE and Standby sleep modes if REF_CLK or GRXCK is running.

All GMAC interrupts can be used to wake up the device from IDLE sleep mode.

In Standby sleep mode, only the Wake On LAN (WOL) interrupt can wake up the CPU, and the corresponding ISR flags will not be set.

**Related Links**
PM – Power Manager

### 24.5.3. Clocks

The GMAC peripheral relies on a system clock from the Main Clock Controller (MCLK) for register access and GMAC MCK.

In MII mode, the actual Transmit or Reference Clock (GTXCK) and Receive Clock (GRXCK) are external signals.

In RMII mode, the actual Reference Clock (REF_CLK) is an external signal.

The respective pins are configured in the PORT peripheral.

**Related Links**
I/O Multiplexing and Considerations
PORT - I/O Pin Controller

### 24.5.4. Interrupt Sources

The GMAC interrupt line is connected to the interrupt controller. Using the GMAC interrupt requires to configure the interrupt controller first.

**Related Links**
Nested Vector Interrupt Controller

### 24.5.5. Events

The event GMAC Timestamp Comparison is connected to the Event System.

**Related Links**
Event System (EVSYS)

## 24.6. Functional Description

### 24.6.1. Media Access Controller

The Transmit Block of the Media Access Controller (MAC) takes data from FIFO, adds preamble, checks and adds padding and frame check sequence (FCS). Both half duplex and full duplex Ethernet modes of operation are supported.

When operating in half duplex mode, the MAC Transmit Block generates data according to the Carrier Sense Multiple Access with Collision Detect (CSMA/CD) protocol. The start of transmission is deferred if Carrier Sense (CRS) is active. If Collision (COL) is detected during transmission, a jam sequence is asserted and the transmission is retried after a random back off. The CRS and COL signals have no effect in full duplex mode.

The Receive Block of the MAC checks for valid preamble, FCS, alignment/length, and presents received frames to the MAC address checking block and FIFO. Software can configure the GMAC


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

to receive jumbo frames of up to 10240 Bytes. It can optionally strip CRC (Cyclic Redundancy Check) from the received frame before transferring it to FIFO.

The Address Checker recognizes four specific 48-bit addresses, can recognize four different types of ID values, and contains a 64-bit Hash register for matching multicast and unicast addresses as required. It can recognize the broadcast address *all-'1'* (0xFFFFFFFFFFFF) and copy all frames. The MAC can also reject all frames that are not VLAN tagged, and recognize Wake on LAN (WOL) events.

The MAC Receive Block supports offloading of IP, TCP, and UDP checksum calculations (both IPv4 and IPv6 packet types supported), and can automatically discard bad checksum frames.

### 24.6.2. IEEE 1588 Time Stamp Unit

The IEEE 1588 time stamp unit (TSU) is implemented as a 94-bit timer.

- The 48 upper bits [93:46] of the timer count seconds and are accessible in the GMAC 1588 Timer Seconds High Register (TSH) and GMAC 1588 Timer Seconds Low Register (TSL).
- The 30 lower bits [45:16] of the timer count nanoseconds and are accessible in the GMAC 1588 Timer Nanoseconds Register (TN).
- The lowest 16 bits [15:0] of the timer count sub-nanoseconds.

The 46 lower bits roll over when they have counted to 1s. The timer increments by a programmable period (to approximately 15.2 femtosecond resolution) with each MCK period and can also be adjusted in 1ns resolution (incremented or decremented) through APB register accesses.

### 24.6.3. AHB Direct Memory Access Interface

The GMAC DMA controller is connected to the MAC FIFO interface and provides a scatter-gather type capability for packet data storage.

The DMA implements packet buffering where dual-port memories are used to buffer multiple frames.

#### 24.6.3.1. Packet Buffer DMA

- Easier to guarantee maximum line rate due to the ability to store multiple frames in the packet buffer, where the number of frames is limited by the amount of packet buffer memory and Ethernet frame size
- Full store and forward, or partial store and forward programmable options (partial store will cater for shorter latency requirements)
- Support for Transmit TCP/IP checksum offload
- Support for priority queuing
- When a collision on the line occurs during transmission, the packet will be automatically replayed directly from the packet buffer memory rather than having to re-fetch through the AHB (full store and forward ONLY)
- Received erroneous packets are automatically dropped before any of the packet is presented to the AHB (full store and forward ONLY), thus reducing AHB activity
- Supports manual RX packet flush capabilities
- Optional RX packet flush when there is lack of AHB resource

#### 24.6.3.2. Partial Store and Forward Using Packet Buffer DMA

The DMA uses SRAM-based packet buffers, and can be programmed into a low latency mode, known as Partial Store and Forward. This mode allows for a reduced latency as the full packet is not buffered before forwarding.
**Note:** This option is only available when the device is configured for full duplex operation.

This feature is enabled via the programmable TX and RX Partial Store and Forward registers (TPSF and RPSF). When the transmit Partial Store and Forward mode is activated, the transmitter will only


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

begin to forward the packet to the MAC when there is enough packet data stored in the packet buffer. Likewise, when the receive Partial Store and Forward mode is activated, the receiver will only begin to forward the packet to the AHB when enough packet data is stored in the packet buffer. The amount of packet data required to activate the forwarding process is programmable via watermark registers. These registers are located at the same address as the partial store and forward enable bits.
**Note:** The minimum operational value for the TX partial store and forward watermark is 20. There is no operational limit for the RX partial store and forward watermark.

Enabling Partial Store and Forward is a useful means to reduce latency, but there are performance implications. The GMAC DMA uses separate transmit and receive lists of buffer descriptors, with each descriptor describing a buffer area in memory. This allows Ethernet packets to be broken up and scattered around the AHB memory space.

#### 24.6.3.3. Receive AHB Buffers

Received frames, optionally including FCS, are written to receive AHB buffers stored in memory. The receive buffer depth is programmable in the range of 64 Bytes to 16 KBytes through the DMA Configuration register (DCFGR), with the default being 128 Bytes.

The start location for each receive AHB buffer is stored in memory in a list of receive buffer descriptors at an address location pointed to by the receive buffer queue pointer. The base address for the receive buffer queue pointer is configured in software using the Receive Buffer Queue Base Address register (RBQB).

Each list entry consists of two words. The first is the address of the receive AHB buffer and the second the receive status.

If the length of a receive frame exceeds the AHB buffer length, the status word for the used buffer is written with zeroes except for the “Start of Frame” bit, which is always set for the first buffer in a frame.

Bit zero of the address field is written to 1 to show that the buffer has been used. The receive buffer manager then reads the location of the next receive AHB buffer and fills that with the next part of the received frame data. AHB buffers are filled until the frame is complete and the final buffer descriptor status word contains the complete frame status. See the following table for details of the receive buffer descriptor list.

**Table 24-2.** Receive Buffer Descriptor Entry

| Bit | Function |
|---|---|
| Word 0 | |
| 31:2 | Address of beginning of buffer |
| 1 | Wrap—marks last descriptor in receive buffer descriptor list. |
| 0 | Ownership—needs to be zero for the GMAC to write data to the receive buffer. The GMAC sets this to one once it has successfully written a frame to memory.<br>Software has to clear this bit before the buffer can be used again. |
| Word 1 | |
| 31 | Global all ones broadcast address detected |
| 30 | Multicast hash match |
| 29 | Unicast hash match |
| 28 | – |
| 27 | Specific Address Register match found, bit 25 and bit 26 indicate which Specific Address Register causes the match. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 24-2.** Receive Buffer Descriptor Entry (continued)

| Bit | Function |
|---|---|
| 26:25 | Specific Address Register match. Encoded as follows:<br>00: Specific Address Register 1 match<br>01: Specific Address Register 2 match<br>10: Specific Address Register 3 match<br>11: Specific Address Register 4 match<br>If more than one specific address is matched only one is indicated with priority 4 down to 1. |
| 24 | This bit has a different meaning depending on whether RX checksum offloading is enabled.<br>With RX checksum offloading disabled: (bit 24 clear in Network Configuration Register)<br>Type ID register match found, bit 22 and bit 23 indicate which type ID register causes the match.<br>With RX checksum offloading enabled: (bit 24 set in Network Configuration Register)<br>0: The frame was not SNAP encoded and/or had a VLAN tag with the Canonical Format Indicator (CFI) bit set.<br>1: The frame was SNAP encoded and had either no VLAN tag or a VLAN tag with the CFI bit not set. |
| 23:22 | This bit has a different meaning depending on whether RX checksum offloading is enabled.<br>With RX checksum offloading disabled: (bit 24 clear in Network Configuration)<br>Type ID register match. Encoded as follows:<br>00: Type ID register 1 match<br>01: Type ID register 2 match<br>10: Type ID register 3 match<br>11: Type ID register 4 match<br>If more than one Type ID is matched only one is indicated with priority 4 down to 1.<br>With RX checksum offloading enabled: (bit 24 set in Network Configuration Register)<br>00: Neither the IP header checksum nor the TCP/UDP checksum was checked.<br>01: The IP header checksum was checked and was correct. Neither the TCP nor UDP checksum was checked.<br>10: Both the IP header and TCP checksum were checked and were correct.<br>11: Both the IP header and UDP checksum were checked and were correct. |
| 21 | VLAN tag detected—type ID of 0x8100. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag has a type ID of 0x8100 |
| 20 | Priority tag detected—type ID of 0x8100 and null VLAN identifier. For packets incorporating the stacked VLAN processing feature, this bit will be set if the second VLAN tag has a type ID of 0x8100 and a null VLAN identifier. |
| 19:17 | VLAN priority—only valid if bit 21 is set. |
| 16 | Canonical format indicator (CFI) bit (only valid if bit 21 is set). |
| 15 | End of frame—when set the buffer contains the end of a frame. If end of frame is not set, then the only valid status bit is start of frame (bit 14). |
| 14 | Start of frame—when set the buffer contains the start of a frame. If both bits 15 and 14 are set, the buffer contains a whole frame. |
| 13 | This bit has a different meaning depending on whether jumbo frames and ignore FCS modes are enabled. If neither mode is enabled this bit will be zero.<br>With jumbo frame mode enabled: (bit 3 set in Network Configuration Register) Additional bit for length of frame (bit[13]), that is concatenated with bits[12:0]<br>With ignore FCS mode enabled and jumbo frames disabled: (bit 26 set in Network Configuration Register and bit 3 clear in Network Configuration Register) This indicates per frame FCS status as follows:<br>0: Frame had good FCS<br>1: Frame had bad FCS, but was copied to memory as ignore FCS enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 24-2.** Receive Buffer Descriptor Entry (continued)

| Bit | Function |
|---|---|
| 12:0 | These bits represent the length of the received frame which may or may not include FCS depending on whether FCS discard mode is enabled.<br>With FCS discard mode disabled: (bit 17 clear in Network Configuration Register)<br>Least significant 12 bits for length of frame including FCS. If jumbo frames are enabled, these 12 bits are concatenated with bit[13] of the descriptor above.<br>With FCS discard mode enabled: (bit 17 set in Network Configuration Register)<br>Least significant 12 bits for length of frame excluding FCS. If jumbo frames are enabled, these 12 bits are concatenated with bit[13] of the descriptor above. |

Each receive AHB buffer start location is a word address. The start of the first AHB buffer in a frame can be offset by up to three Bytes, depending on the value written to bits 14 and 15 of the Network Configuration register (NCFGR). If the start location of the AHB buffer is offset, the available length of the first AHB buffer is reduced by the corresponding number of Bytes.

To receive frames, the AHB buffer descriptors must be initialized by writing an appropriate address to bits 31:2 in the first word of each list entry. Bit 0 must be written with zero. Bit 1 is the wrap bit and indicates the last entry in the buffer descriptor list.

The start location of the receive buffer descriptor list must be written with the receive buffer queue base address before reception is enabled (receive enable in the Network Control register NCR). Once reception is enabled, any writes to the Receive Buffer Queue Base Address register (RBQB) are ignored. When read, it will return the current pointer position in the descriptor list, though this is only valid and stable when receive is disabled.

If the filter block indicates that a frame should be copied to memory, the receive data DMA operation starts writing data into the receive buffer. If an error occurs, the buffer is recovered.

An internal counter within the GMAC represents the receive buffer queue pointer and it is not visible through the CPU interface. The receive buffer queue pointer increments by two words after each buffer has been used. It re-initializes to the receive buffer queue base address if any descriptor has its wrap bit set.

As receive AHB buffers are used, the receive AHB buffer manager sets bit zero of the first word of the descriptor to logic one indicating the AHB buffer has been used.

Software should search through the “used” bits in the AHB buffer descriptors to find out how many frames have been received, checking the start of frame and end of frame bits.

When the DMA is configured in the packet buffer Partial Store And Forward mode, received frames are written out to the AHB buffers as soon as enough frame data exists in the packet buffer. For both cases, this may mean several full AHB buffers are used before some error conditions can be detected. If a receive error is detected the receive buffer currently being written will be recovered. Previous buffers will not be recovered. As an example, when receiving frames with cyclic redundancy check (CRC) errors or excessive length, it is possible that a frame fragment might be stored in a sequence of AHB receive buffers. Software can detect this by looking for start of frame bit set in a buffer following a buffer with no end of frame bit set.

To function properly, a 10/100 Ethernet system should have no excessive length frames or frames greater than 128 Bytes with CRC errors. Collision fragments will be less than 128 Bytes long, therefore it will be a rare occurrence to find a frame fragment in a receive AHB buffer, when using the default value of 128 Bytes for the receive buffers size.

When in packet buffer full store and forward mode, only good received frames are written out of the DMA, so no fragments will exist in the AHB buffers due to MAC receiver errors. There is still the possibility of fragments due to DMA errors, for example used bit read on the second buffer of a multi-buffer frame.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

If bit zero of the receive buffer descriptor is already set when the receive buffer manager reads the location of the receive AHB buffer, the buffer has been already used and cannot be used again until software has processed the frame and cleared bit zero. In this case, the “buffer not available” bit in the receive status register is set and an interrupt triggered. The receive resource error statistics register is also incremented.

When the DMA is configured in the packet buffer full store and forward mode, the user can optionally select whether received frames should be automatically discarded when no AHB buffer resource is available. This feature is selected via the DMA Discard Receive Packets bit in the DMA Configuration register (DCFGR.DDRP). By default, the received frames are not automatically discarded. If this feature is off, then received packets will remain to be stored in the SRAM-based packet buffer until AHB buffer resource next becomes available. This may lead to an eventual packet buffer overflow if packets continue to be received when bit zero (used bit) of the receive buffer descriptor remains set.
**Note:** After a used bit has been read, the receive buffer manager will re-read the location of the receive buffer descriptor every time a new packet is received. When the DMA is not configured in the packet buffer full store and forward mode and a used bit is read, the frame currently being received will be automatically discarded.

When the DMA is configured in the packet buffer full store and forward mode, a receive overrun condition occurs when the receive SRAM-based packet buffer is full, or because HRESP was not OK. In all other modes, a receive overrun condition occurs when either the AHB bus was not granted quickly enough, or because HRESP was not OK, or because a new frame has been detected by the receive block, but the status update or write back for the previous frame has not yet finished. For a receive overrun condition, the receive overrun interrupt is asserted and the buffer currently being written is recovered. The next frame that is received whose address is recognized reuses the buffer.

In any packet buffer mode, writing a '1' to the Flush Next Package bit in the NCR register (NCR.FNP) will force a packet from the external SRAM-based receive packet buffer to be flushed. This feature is only acted upon when the RX DMA is not currently writing packet data out to AHB, i.e., it is in an IDLE state. If the RX DMA is active, NCR.FNP=1 is ignored.

#### 24.6.3.4. Transmit AHB Buffers

Frames to transmit are stored in one or more transmit AHB buffers. Transmit frames can be between 1 and 16384 Bytes long, so it is possible to transmit frames longer than the maximum length specified in the IEEE 802.3 standard. It should be noted that zero length AHB buffers are allowed and that the maximum number of buffers permitted for each transmit frame is 128.

The start location for each transmit AHB buffer is stored in memory in a list of transmit buffer descriptors at a location pointed to by the transmit buffer queue pointer. The base address for this queue pointer is set in software using the Transmit Buffer Queue Base Address register. Each list entry consists of two words. The first is the Byte address of the transmit buffer and the second containing the transmit control and status. For the packet buffer DMA, the start location for each AHB buffer is a Byte address, the bottom bits of the address being used to offset the start of the data from the data-word boundary (i.e., bits 2,1 and 0 are used to offset the address for 64-bit data paths).

Frames can be transmitted with or without automatic Cyclic Redundancy Checksum (CRC) generation. If CRC is automatically generated, pad will also be automatically generated to take frames to a minimum length of 64 Bytes. When CRC is not automatically generated (as defined in word 1 of the transmit buffer descriptor), the frame is assumed to be at least 64 Bytes long and pad is not generated.

An entry in the transmit buffer descriptor list is described in this table:

**Table 24-3.** Transmit Buffer Descriptor Entry

| Bit | Function |
|---|---|
| Word 0 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 24-3.** Transmit Buffer Descriptor Entry (continued)

| Bit | Function |
|---|---|
| 31:0 | Byte address of buffer |
| Word 1 | |
| 31 | Used—must be zero for the GMAC to read data to the transmit buffer. The GMAC sets this to one for the first buffer of a frame once it has been successfully transmitted. Software must clear this bit before the buffer can be used again. |
| 30 | Wrap—marks last descriptor in transmit buffer descriptor list. This can be set for any buffer within the frame. |
| 29 | Retry limit exceeded, transmit error detected |
| 28 | Reserved. |
| 27 | Transmit frame corruption due to AHB error—set if an error occurs while midway through reading transmit frame from the AHB, including HRESP errors and buffers exhausted mid frame (if the buffers run out during transmission of a frame then transmission stops, FCS shall be bad and GTXER asserted).<br>Also set if single frame is too large for configured packet buffer memory size. |
| 26 | Late collision, transmit error detected. |
| 25:23 | Reserved |
| 22:20 | Transmit IP/TCP/UDP checksum generation offload errors:<br>000: No Error.<br>001: The Packet was identified as a VLAN type, but the header was not fully complete, or had an error in it.<br>010: The Packet was identified as a SNAP type, but the header was not fully complete, or had an error in it.<br>011: The Packet was not of an IP type, or the IP packet was invalidly short, or the IP was not of type IPv4/IPv6.<br>100: The Packet was not identified as VLAN, SNAP or IP.<br>101: Non supported packet fragmentation occurred. For IPv4 packets, the IP checksum was generated and inserted.<br>110: Packet type detected was not TCP or UDP. TCP/UDP checksum was therefore not generated. For IPv4 packets, the IP checksum was generated and inserted.<br>111: A premature end of packet was detected and the TCP/UDP checksum could not be generated. |
| 19:17 | Reserved |
| 16 | No CRC to be appended by MAC. When set, this implies that the data in the buffers already contains a valid CRC, hence no CRC or padding is to be appended to the current frame by the MAC.<br>This control bit must be set for the first buffer in a frame and will be ignored for the subsequent buffers of a frame.<br>Note that this bit must be clear when using the transmit IP/TCP/UDP checksum generation offload, otherwise checksum generation and substitution will not occur. |
| 15 | Last buffer, when set this bit will indicate the last buffer in the current frame has been reached. |
| 14 | Reserved |
| 13:0 | Length of buffer |

To transmit frames, the buffer descriptors must be initialized by writing an appropriate Byte address to bits [31:0] of the first word of each descriptor list entry.

The second word of the transmit buffer descriptor is initialized with control information that indicates the length of the frame, whether or not the MAC is to append CRC and whether the buffer is the last buffer in the frame.

After transmission the status bits are written back to the second word of the first buffer along with the used bit. Bit 31 is the used bit which must be zero when the control word is read if transmission is to take place. It is written to '1' once the frame has been transmitted. Bits[29:20] indicate various transmit error conditions. Bit 30 is the wrap bit which can be set for any buffer within a frame. If no wrap bit is encountered the queue pointer continues to increment.

The Transmit Buffer Queue Base Address register can only be updated while transmission is disabled or halted; otherwise any attempted write will be ignored. When transmission is halted the transmit buffer queue pointer will maintain its value. Therefore when transmission is restarted the next descriptor read from the queue will be from immediately after the last successfully transmitted frame. As long as transmit is disabled by writing a '0' to the Transmit Enable bit in the Network


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Control register (NCR.TXEN), the transmit buffer queue pointer resets to point to the address indicated by the Transmit Buffer Queue Base Address register (TBQB).
**Note:** Disabling receive does not have the same effect on the receive buffer queue pointer.

Once the transmit queue is initialized, transmit is activated by writing a '1' to the Start Transmission bit of the Network Control register (NCR.TSTART). Transmit is halted when a buffer descriptor with its used bit set is read, a transmit error occurs, or by writing to the Transmit Halt bit of the Network Control register (NCR.THALT). Transmission is suspended if a pause frame is received while the Transmit Pause Frame bit is '1' in the Network Configuration register (NCR.TXPF). Rewriting the Start bit (NCR.TSTART) while transmission is active is allowed. This is implemented by the Transmit Go variable which is readable in the Transmit Status register (TSR.TXGO). The TXGO variable is reset when:

- Transmit is disabled
- A buffer descriptor with its used bit set, is read
- Bit 10, THALT, of the Network Control register is written
- There is a transmit error such as too many retries or a transmit underrun

To set TXGO, write a '1' to NCR.TSTART. Transmit halt does not take effect until any ongoing transmit finishes.

If the DMA is configured for packet buffer Partial Store and Forward mode and a collision occurs during transmission of a multi-buffer frame, transmission will automatically restart from the first buffer of the frame. For packet buffer mode, the entire contents of the frame are read into the transmit packet buffer memory, so the retry attempt will be replayed directly from the packet buffer memory rather than having to re-fetch through the AHB.

If a used bit is read midway through transmission of a multi-buffer frame, this is treated as a transmit error. Transmission stops, GTXER is asserted and the FCS will be bad.

If transmission stops due to a transmit error or a used bit being read, transmission restarts from the first buffer descriptor of the frame being transmitted when the transmit start bit is rewritten.

#### 24.6.3.5. DMA Bursting on the AHB

The DMA will always use SINGLE, or INCR type AHB accesses for buffer management operations. When performing data transfers, the AHB burst length is selected by the Fixed Burst Length for DMA Data Operations bit field in the DMA Configuration register (DCFGR.FBLDO) so that either SINGLEor fixed length incrementing bursts (INCR4, INCR8 or INCR16) are used where possible:

When there is enough space and enough data to be transferred, the programmed fixed length bursts will be used. If there is not enough data or space available, for example when at the beginning or the end of a buffer, SINGLE type accesses are used. Also SINGLE type accesses are used at 1024 Byte boundaries, so that the 1 KByte boundaries are not burst over as per AHB requirements.

The DMA will not terminate a fixed length burst early, unless an error condition occurs on the AHB or if receive or transmit are disabled in the Network Control register (NCR).

#### 24.6.3.6. DMA Packet Buffer

The DMA uses packet buffers for both transmit and receive paths. This mode allows multiple packets to be buffered in both transmit and receive directions. This allows the DMA to withstand far greater access latencies on the AHB and make more efficient use of the AHB bandwidth. There are two modes of operation—Full Store and Forward and Partial Store and Forward.

As described above, the DMA can be programmed into a low latency mode, known as Partial Store and Forward. For further details of this mode, see the related Links.

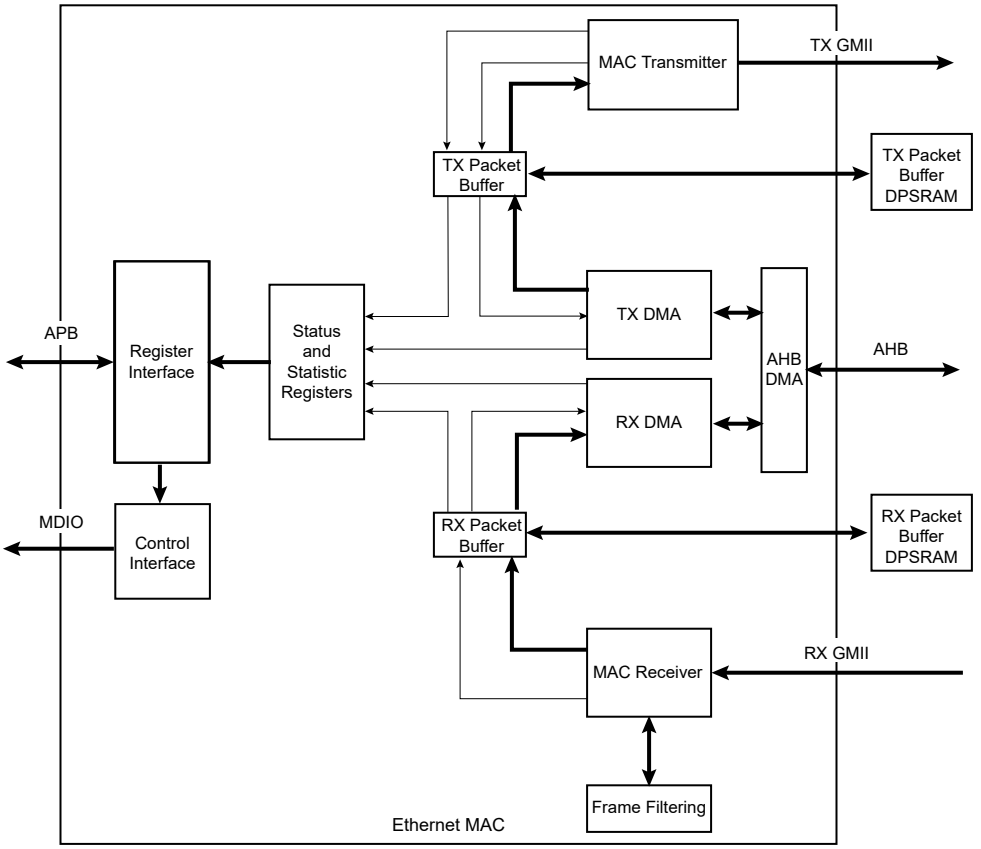
When the DMA is in full store and forward mode, full packets are buffered which provides the possibility to:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Discard packets with error on the receive path before they are partially written out of the DMA, thus saving AHB bus bandwidth and driver processing overhead,
- Retry collided transmit frames from the buffer, thus saving AHB bus bandwidth,
- Implement transmit IP/TCP/UDP checksum generation offload.

With the packet buffers included, the structure of the GMAC data paths is shown in this image:

**Figure 24-2.** Data Paths with Packet Buffers Included




### 24.6.3.7. Transmit Packet Buffer

The transmitter packet buffer will continue attempting to fetch frame data from the AHB system memory until the packet buffer itself is full, at which point it will attempt to maintain its full level.

To accommodate the status and statistics associated with each frame, three words per packet (or two if the GMAC is configured in 64-bit data path mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet. Storing the status in the DPRAM is required in order to decouple the DMA interface of the buffer from the MAC interface, to update the MAC status/statistics and to generate interrupts in the order in which the packets that they represent were fetched from the AHB memory.

If any errors occur on the AHB while reading the transmit frame, the fetching of packet data from AHB memory is halted. The MAC transmitter will continue to fetch packet data, thereby emptying the packet buffer and allowing any good (non-erroneous) frames to be transmitted successfully. Once these have been fully transmitted, the status/statistics for the erroneous frame will be updated and software will be informed via an interrupt that an AHB error occurred. This way, the error is reported in the correct packet order.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The transmit packet buffer will only attempt to read more frame data from the AHB when space is available in the packet buffer memory. If space is not available it must wait until the a packet fetched by the MAC completes transmission and is subsequently removed from the packet buffer memory.
**Note:** If full store and forward mode is active and if a single frame is fetched that is too large for the packet buffer memory, the frame is flushed and the DMA halted with an error status. This is because a complete frame must be written into the packet buffer before transmission can begin, and therefore the minimum packet buffer memory size should be chosen to satisfy the maximum frame to be transmitted in the application.

In full store and forward mode, once the complete transmit frame is written into the packet buffer memory, a trigger is sent across to the MAC transmitter, which will then begin reading the frame from the packet buffer memory. Since the whole frame is present and stable in the packet buffer memory an underflow of the transmitter is not possible. The frame is kept in the packet buffer until notification is received from the MAC that the frame data has either been successfully transmitted or can no longer be retransmitted (too many retries in half duplex mode). When this notification is received the frame is flushed from memory to make room for a new frame to be fetched from AHB system memory.

In Partial Store and Forward mode, a trigger is sent across to the MAC transmitter as soon as sufficient packet data is available, which will then begin fetching the frame from the packet buffer memory. If, after this point, the MAC transmitter is able to fetch data from the packet buffer faster than the AHB DMA can fill it, an underflow of the transmitter is possible. In this case, the transmission is terminated early, and the packet buffer is completely flushed. Transmission can only be restarted by writing a '1' to the Transmit Start bit in the Network Control register (NCR.TSTART).

In half duplex mode, the frame is kept in the packet buffer until notification is received from the MAC that the frame data has either been successfully transmitted or can no longer be retransmitted (too many retries in half duplex mode). When this notification is received the frame is flushed from memory to make room for a new frame to be fetched from AHB system memory.

In full duplex mode, the frame is removed from the packet buffer on the fly.

Other than underflow, the only MAC related errors that can occur are due to collisions during half duplex transmissions. When a collision occurs the frame still exists in the packet buffer memory so can be retried directly from there. After sixteen failed transmit attempts, the frame will be flushed from the packet buffer.

#### 24.6.3.8. Receive Packet Buffer

The receive packet buffer stores frames from the MAC receiver along with their status and statistics. Frames with errors are flushed from the packet buffer memory, while good frames are pushed onto the DMA AHB interface.

The receiver packet buffer monitors the FIFO write interface from the MAC receiver and translates the FIFO pushes into packet buffer writes. At the end of the received frame the status and statistics are buffered so that the information can be used when the frame is read out. When programmed in full store and forward mode and the frame has an error, the frame data is immediately flushed from the packet buffer memory allowing subsequent frames to utilize the freed up space. The status and statistics for bad frames are still used to update the GMAC registers.

To accommodate the status and statistics associated with each frame, three words per packet (or two if configured in 64-bit datapath mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet.

The receiver packet buffer will also detect a full condition so that an overflow condition can be detected. If this occurs, subsequent packets are dropped and an RX overflow interrupt is raised.

For full store and forward, the DMA only begins packet fetches once the status and statistics for a frame are available. If the frame has a bad status due to a frame error, the status and statistics are passed on to the GMAC registers. If the frame has a good status, the information is used to


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

read the frame from the packet buffer memory and burst onto the AHB using the DMA buffer management protocol. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

If Partial Store and Forward mode is active, the DMA will begin fetching the packet data before the status is available. As soon as the status becomes available, the DMA will fetch this information as soon as possible before continuing to fetch the remainder of the frame. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

### 24.6.4. MAC Transmit Block

The MAC transmitter can operate in either half duplex or full duplex mode and transmits frames in accordance with the Ethernet IEEE 802.3 standard. In half duplex mode, the CSMA/CD protocol of the IEEE 802.3 specification is followed.

A small input buffer receives data through the FIFO interface which will extract data in 32-bit form. All subsequent processing prior to the final output is performed in bytes.

Transmit data can be output using the MII interface.

Frame assembly starts by adding preamble and the start frame delimiter. Data is taken from the transmit FIFO interface a word at a time.

If necessary, padding is added to take the frame length to 60 bytes. CRC is calculated using an order 32-bit polynomial. This is inverted and appended to the end of the frame taking the frame length to a minimum of 64 bytes. If the no CRC bit is set in the second word of the last buffer descriptor of a transmit frame, neither pad nor CRC are appended. The no CRC bit can also be set through the FIFO interface.

In full duplex mode (at all data rates), frames are transmitted immediately. Back to back frames are transmitted at least 96 bit times apart to guarantee the interframe gap.

In half duplex mode, the transmitter checks carrier sense. If asserted, the transmitter waits for the signal to become inactive, and then starts transmission after the interframe gap of 96 bit times. If the collision signal is asserted during transmission, the transmitter will transmit a jam sequence of 32 bits taken from the data register and then retry transmission after the back off time has elapsed. If the collision occurs during either the preamble or Start Frame Delimiter (SFD), then these fields will be completed prior to generation of the jam sequence.

The back off time is based on an XOR of the 10 least significant bits of the data coming from the transmit FIFO interface and a 10-bit pseudo random number generator. The number of bits used depends on the number of collisions seen. After the first collision 1 bit is used, then the second 2 bits and so on up to the maximum of 10 bits. All 10 bits are used above ten collisions. An error will be indicated and no further attempts will be made if 16 consecutive attempts cause collision. This operation is compliant with the description in Clause 4.2.3.2.5 of the IEEE 802.3 standard which refers to the truncated binary exponential back off algorithm.

In 10/100 mode, both collisions and late collisions are treated identically, and back off and retry will be performed up to 16 times. This condition is reported in the transmit buffer descriptor word 1 (late collision, bit 26) and also in the Transmit Status register (late collision, bit 7). An interrupt can also be generated (if enabled) when this exception occurs, and bit 5 in the Interrupt Status register will be set.

In all modes of operation, if the transmit DMA underruns, a bad CRC is automatically appended using the same mechanism as jam insertion and the GTXER signal is asserted. For a properly configured system this should never happen and also it is impossible if configured to use the DMA with packet buffers, as the complete frame is buffered in local packet buffer memory.

By setting when bit 28 is set in the Network Configuration register, the Inter Packet Gap (IPG) may be stretched beyond 96 bits depending on the length of the previously transmitted frame and the


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

value written to the IPG Stretch register (IPGS). The least significant 8 bits of the IPG Stretch register multiply the previous frame length (including preamble). The next significant 8 bits (+1 so as not to get a divide by zero) divide the frame length to generate the IPG. IPG stretch only works in full duplex mode and when bit 28 is set in the Network Configuration register. The IPG Stretch register cannot be used to shrink the IPG below 96 bits.

If the back pressure bit is set in the Network Control register, or if the HDFC configuration bit is set in the UR register (10M or 100M half duplex mode), the transmit block transmits 64 bits of data, which can consist of 16 nibbles of 1011 or in bit rate mode 64 1s, whenever it sees an incoming frame to force a collision. This provides a way of implementing flow control in half duplex mode.

### 24.6.5. MAC Receive Block

All processing within the MAC receive block is implemented using a 16-bit data path. The MAC receive block checks for valid preamble, FCS, alignment and length, presents received frames to the FIFO interface and stores the frame destination address for use by the address checking block.

If, during the frame reception, the frame is found to be too long, a bad frame indication is sent to the FIFO interface. The receiver logic ceases to send data to memory as soon as this condition occurs.

At end of frame reception the receive block indicates to the DMA block whether the frame is good or bad. The DMA block will recover the current receive buffer if the frame was bad.

Ethernet frames are normally stored in DMA memory complete with the FCS. Setting the FCS remove bit in the network configuration (bit 17) causes frames to be stored without their corresponding FCS. The reported frame length field is reduced by four bytes to reflect this operation.

The receive block signals to the register block to increment the alignment, CRC (FCS), short frame, long frame, jabber or receive symbol errors when any of these exception conditions occur.

If bit 26 is set in the network configuration, CRC errors will be ignored and CRC errored frames will not be discarded, though the Frame Check Sequence Errors statistic register will still be incremented. Additionally, if not enabled for jumbo frames mode, then bit[13] of the receiver descriptor word 1 will be updated to indicate the FCS validity for the particular frame. This is useful for applications such as EtherCAT whereby individual frames with FCS errors must be identified.

Received frames can be checked for length field error by setting the length field error frame discard bit of the Network Configuration register (bit-16). When this bit is set, the receiver compares a frame's measured length with the length field (bytes 13 and 14) extracted from the frame. The frame is discarded if the measured length is shorter. This checking procedure is for received frames between 64 bytes and 1518 bytes in length.

Each discarded frame is counted in the 10-bit length field error statistics register. Frames where the length field is greater than or equal to 0x0600 hex will not be checked.

### 24.6.6. Checksum Offload for IP, TCP and UDP

The GMAC can be programmed to perform IP, TCP and UDP checksum offloading in both receive and transmit directions, which is enabled by setting bit 24 in the Network Configuration register for receive and bit 11 in the DMA Configuration register for transmit.

IPv4 packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header. TCP and UDP packets contain a 16-bit checksum field, which is the 16-bit 1’s complement of the 1’s complement sum of all 16-bit words in the header, the data and a conceptual IP pseudo header.

To calculate these checksums in software requires each byte of the packet to be processed. For TCP and UDP this can use a large amount of processing power. Offloading the checksum calculation to hardware can result in significant performance improvements.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For IP, TCP or UDP checksum offload to be useful, the operating system containing the protocol stack must be aware that this offload is available so that it can make use of the fact that the hardware can either generate or verify the checksum.

#### 24.6.6.1. Receiver Checksum Offload

When receive checksum offloading is enabled in the GMAC Network Configuration Register (NCFGR.RXCOEN), the IPv4 header checksum is checked as per RFC 791, where the packet meets the following criteria:

- If present, the VLAN header must be four octets long and the CFI bit must not be set.
- Encapsulation must be RFC 894 Ethernet Type Encoding or RFC 1042 SNAP Encoding.
- IPv4 packet
- IP header is of a valid length

The GMAC also checks the TCP checksum as per RFC 793, or the UDP checksum as per RFC 768, if the following criteria are met:

- IPv4 or IPv6 packet
- Good IP header checksum (if IPv4)
- No IP fragmentation
- TCP or UDP packet

When an IP, TCP or UDP frame is received, the receive buffer descriptor gives an indication if the GMAC was able to verify the checksums. There is also an indication if the frame had SNAP encapsulation. These indication bits will replace the type ID match indication bits when the receive checksum offload is enabled. For details of these indication bits refer to "Receive Buffer Descriptor Entry".

If any of the checksums are verified as incorrect by the GMAC, the packet is discarded and the appropriate statistics counter incremented.

#### 24.6.6.2. Transmitter Checksum Offload

The transmitter checksum offload is only available if the full store and forward mode is enabled. This is because the complete frame to be transmitted must be read into the packet buffer memory before the checksum can be calculated and written back into the headers at the beginning of the frame.

Transmitter checksum offload is enabled by setting bit [11] in the DMA Configuration register. When enabled, it will monitor the frame as it is written into the transmitter packet buffer memory to automatically detect the protocol of the frame. Protocol support is identical to the receiver checksum offload.

For transmit checksum generation and substitution to occur, the protocol of the frame must be recognized and the frame must be provided without the FCS field, by making sure that bit [16] of the transmit descriptor word 1 is clear. If the frame data already had the FCS field, this would be corrupted by the substitution of the new checksum fields.

If these conditions are met, the transmit checksum offload engine will calculate the IP, TCP and UDP checksums as appropriate. Once the full packet is completely written into packet buffer memory, the checksums will be valid and the relevant DPRAM locations will be updated for the new checksum fields as per standard IP/TCP and UDP packet structures.

If the transmitter checksum engine is prevented from generating the relevant checksums, bits [22:20] of the transmitter DMA writeback status will be updated to identify the reason for the error. Note that the frame will still be transmitted but without the checksum substitution, as typically the reason that the substitution did not occur was that the protocol was not recognized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.6.7. MAC Filtering Block

The filter block determines which frames should be written to the FIFO interface and on to the DMA.

Whether a frame is passed depends on what is enabled in the Network Configuration register, the state of the external matching pins, the contents of the specific address, type and Hash registers and the frame's destination address and type field.

If bit 25 of the Network Configuration register is not set, a frame will not be copied to memory if the GMAC is transmitting in half duplex mode at the time a destination address is received.

Ethernet frames are transmitted a byte at a time, least significant bit first. The first six bytes (48 bits) of an Ethernet frame make up the destination address. The first bit of the destination address, which is the LSB of the first byte of the frame, is the group or individual bit. This is one for multicast addresses and zero for unicast. The all ones address is the broadcast address and a special case of multicast.

The GMAC supports recognition of four specific addresses. Each specific address requires two registers, Specific Address register Bottom and Specific Address register Top. Specific Address register Bottom stores the first four bytes of the destination address and Specific Address register Top contains the last two bytes. The addresses stored can be specific, group, local or universal.

The destination address of received frames is compared against the data stored in the Specific Address registers once they have been activated. The addresses are deactivated at reset or when their corresponding Specific Address register Bottom is written. They are activated when Specific Address register Top is written. If a receive frame address matches an active address, the frame is written to the FIFO interface and on to DMA memory.

Frames may be filtered using the type ID field for matching. Four type ID registers exist in the register address space and each can be enabled for matching by writing a one to the MSB (bit 31) of the respective register. When a frame is received, the matching is implemented as an OR function of the various types of match.

The contents of each type ID register (when enabled) are compared against the length/type ID of the frame being received (e.g., bytes 13 and 14 in non-VLAN and non-SNAP encapsulated frames) and copied to memory if a match is found. The encoded type ID match bits (Word 0, Bit 22 and Bit 23) in the receive buffer descriptor status are set indicating which type ID register generated the match, if the receive checksum offload is disabled.

The reset state of the type ID registers is zero, hence each is initially disabled.

The following example illustrates the use of the address and type ID match registers for a MAC address of 21:43:65:87:A9:CB:

| | |
|---|---|
| Preamble | 55 |
| SFD | D5 |
| DA (Octet 0 - LSB) | 21 |
| DA (Octet 1) | 43 |
| DA (Octet 2) | 65 |
| DA (Octet 3) | 87 |
| DA (Octet 4) | A9 |
| DA (Octet 5 - MSB) | CB |
| SA (LSB) | 00 (see **Note**) |
| SA | 00(see **Note**) |
| SA | 00(see **Note**) |
| SA | 00(see **Note**) |
| SA | 00(see **Note**) |
| SA (MSB) | 00(see **Note**) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| | |
|---|---|
| Type ID (MSB) | 43 |
| Type ID (LSB) | 21 |

**Note:** Contains the address of the transmitting device.

The previous sequence shows the beginning of an Ethernet frame. Byte order of transmission is from top to bottom, as shown. For a successful match to specific address 1, the following address matching registers must be set up:

Specific Address 1 Bottom register (SAB1) (Address 0x088) 0x87654321

Specific Address 1 Top register (SAT1) (Address 0x08C) 0x0000CBA9

For a successful match to the type ID, the following Type ID Match 1 register must be set up:

Type ID Match 1 register (TIDM1) (Address 0x0A8) 0x80004321

### 24.6.8. Broadcast Address

Frames with the broadcast address of 0xFFFFFFFFFFFF are stored to memory only if the 'no broadcast' bit in the Network Configuration register is set to zero.

### 24.6.9. Hash Addressing

The hash address register is 64 bits long and takes up two locations in the memory map. The least significant bits are stored in Hash Register Bottom and the most significant bits in Hash Register Top.

The unicast hash enable and the multicast hash enable bits in the Network Configuration register enable the reception of hash matched frames. The destination address is reduced to a 6-bit index into the 64-bit Hash register using the following hash function: The hash function is an XOR of every sixth bit of the destination address.

hash_index[05] = da[05] ^ da[11] ^ da[17] ^ da[23] ^ da[29] ^ da[35] ^ da[41] ^ da[47]

hash_index[04] = da[04] ^ da[10] ^ da[16] ^ da[22] ^ da[28] ^ da[34] ^ da[40] ^ da[46]

hash_index[03] = da[03] ^ da[09] ^ da[15] ^ da[21] ^ da[27] ^ da[33] ^ da[39] ^ da[45]

hash_index[02] = da[02] ^ da[08] ^ da[14] ^ da[20] ^ da[26] ^ da[32] ^ da[38] ^ da[44]

hash_index[01] = da[01] ^ da[07] ^ da[13] ^ da[19] ^ da[25] ^ da[31] ^ da[37] ^ da[43]

hash_index[00] = da[00] ^ da[06] ^ da[12] ^ da[18] ^ da[24] ^ da[30] ^ da[36] ^ da[42]

da[0] represents the least significant bit of the first byte received, that is, the multicast/unicast indicator, and da[47] represents the most significant bit of the last byte received.

If the hash index points to a bit that is set in the Hash register then the frame will be matched according to whether the frame is multicast or unicast.

A multicast match will be signaled if the multicast hash enable bit is set, da[0] is logic 1 and the hash index points to a bit set in the Hash register.

A unicast match will be signaled if the unicast hash enable bit is set, da[0] is logic 0 and the hash index points to a bit set in the Hash register.

To receive all multicast frames, the Hash register should be set with all ones and the multicast hash enable bit should be set in the Network Configuration register.

### 24.6.10. Copy all Frames (Promiscuous Mode)

If the Copy All Frames bit is set in the Network Configuration register then all frames (except those that are too long, too short, have FCS errors or have GRXER asserted during reception) will be copied to memory. Frames with FCS errors will be copied if bit 26 is set in the Network Configuration register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.6.11. Disable Copy of Pause Frames

Pause frames can be prevented from being written to memory by setting the disable copying of pause frames control bit 23 in the Network Configuration register. When set, pause frames are not copied to memory regardless of the Copy All Frames bit, whether a hash match is found, a type ID match is identified or if a destination address match is found.

### 24.6.12. VLAN Support

The following table describes an Ethernet encoded 802.1Q VLAN tag.

**Table 24-4.** 802.1Q VLAN Tag

| TPID (Tag Protocol Identifier) 16 bits | TCI (Tag Control Information) 16 bits |
|---|---|
| 0x8100 | First 3 bits priority, then CFI bit, last 12 bits VID |

The VLAN tag is inserted at the 13th byte of the frame adding an extra four bytes to the frame. To support these extra four bytes, the GMAC can accept frame lengths up to 1536 bytes by setting bit 8 in the Network Configuration register.

If the VID (VLAN identifier) is null (0x000) this indicates a priority-tagged frame.

The following bits in the receive buffer descriptor status word give information about VLAN tagged frames:-

- Bit 21 set if receive frame is VLAN tagged (i.e., type ID of 0x8100).
- Bit 20 set if receive frame is priority tagged (i.e., type ID of 0x8100 and null VID). (If bit 20 is set, bit 21 will be set also.)
- Bit 19, 18 and 17 set to priority if bit 21 is set.
- Bit 16 set to CFI if bit 21 is set.

The GMAC can be configured to reject all frames except VLAN tagged frames by setting the discard non-VLAN frames bit in the Network Configuration register.

### 24.6.13. Wake on LAN Support

The receive block supports Wake on LAN by detecting the following events on incoming receive frames:

- Magic packet
- Address Resolution Protocol (ARP) request to the device IP address
- Specific address 1 filter match
- Multicast hash filter match

These events can be individually enabled through bits [19:16] of the Wake on LAN register. Also, for Wake on LAN detection to occur, receive enable must be set in the Network Control register, however a receive buffer does not have to be available.

In case of an ARP request, specific address 1 or multicast filter events will occur even if the frame is errored. For magic packet events, the frame must be correctly formed and error free.

A magic packet event is detected if all of the following are true:

- Magic packet events are enabled through bit 16 of the Wake on LAN register
- The frame's destination address matches specific address 1
- The frame is correctly formed with no errors
- The frame contains at least 6 bytes of 0xFF for synchronization
- There are 16 repetitions of the contents of Specific Address 1 register immediately following the synchronization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

An ARP request event is detected if all of the following are true:

- ARP request events are enabled through bit 17 of the Wake on LAN register
- Broadcasts are allowed by bit 5 in the Network Configuration register
- The frame has a broadcast destination address (bytes 1 to 6)
- The frame has a type ID field of 0x0806 (bytes 13 and 14)
- The frame has an ARP operation field of 0x0001 (bytes 21 and 22)
- The least significant 16 bits of the frame's ARP target protocol address (bytes 41 and 42) match the value programmed in bits[15:0] of the Wake on LAN register

The decoding of the ARP fields adjusts automatically if a VLAN tag is detected within the frame. The reserved value of 0x0000 for the Wake on LAN target address value will not cause an ARP request event, even if matched by the frame.

A specific address 1 filter match event will occur if all of the following are true:

- Specific address 1 events are enabled through bit 18 of the Wake on LAN register
- The frame's destination address matches the value programmed in the Specific Address 1 registers

A multicast filter match event will occur if all of the following are true:

- Multicast hash events are enabled through bit 19 of the Wake on LAN register
- Multicast hash filtering is enabled through bit 6 of the Network Configuration register
- The frame destination address matches against the multicast hash filter
- The frame destination address is not a broadcast

### 24.6.14. IEEE 1588 Support

IEEE 1588 is a standard for precision time synchronization in local area networks. It works with the exchange of special Precision Time Protocol (PTP) frames. The PTP messages can be transported over IEEE 802.3/Ethernet, over Internet Protocol Version 4 or over Internet Protocol Version 6 as described in the annex of IEEE P1588.D2.1.

GMAC output pins indicate the message time-stamp point (asserted on the start packet delimiter and de-asserted at end of frame) for all frames and the passage of PTP event frames (asserted when a PTP event frame is detected and de-asserted at end of frame).

IEEE 802.1AS is a subset of IEEE 1588. One difference is that IEEE 802.1AS uses the Ethernet multicast address 0180C200000E for sync frame recognition whereas IEEE 1588 does not. GMAC is designed to recognize sync frames with both IEEE 802.1AS and IEEE 1588 addresses and so can support both 1588 and 802.1AS frame recognition simultaneously.

Synchronization between Host and Client clocks is a two stage process.

First, the offset between the Host and Client clocks is corrected by the Host sending a sync frame to the Client with a follow up frame containing the exact time the sync frame was sent. Hardware assist modules at the Host and Client side detect exactly when the sync frame was sent by the Host and received by the Client. The Client then corrects its clock to match the Host clock.

Second, the transmission delay between the Host and Client is corrected. The Client sends a delay request frame to the Host which sends a delay response frame in reply. Hardware assist modules at the Host and Client side detect exactly when the delay request frame was sent by the Client and received by the Host. The Client will now have enough information to adjust its clock to account for delay. For example, if the Client was assuming zero delay, the actual delay will be half the difference between the transmit and receive time of the delay request frame (assuming equal transmit and receive times) because the Client clock will be lagging the Host clock by the delay time already.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The time-stamp is taken when the message time-stamp point passes the clock time-stamp point. This can generate an interrupt if enabled (IER). However, MAC Filtering configuration is needed to actually 'copy' the message to memory. For Ethernet, the message time-stamp point is the SFD and the clock time-stamp point is the MII interface. (The IEEE 1588 specification refers to sync and delay_req messages as event messages as these require time-stamping. These events are captured in the registers TSSx, EFTx and EFRx, respectively. Follow up, delay response and management messages do not require time-stamping and are referred to as general messages.)

1588 version 2 defines two additional PTP event messages. These are the peer delay request (Pdelay_Req) and peer delay response (Pdelay_Resp) messages. These events are captured in the registers PEFTx and PEFRx, respectively. These messages are used to calculate the delay on a link. Nodes at both ends of a link send both types of frames (regardless of whether they contain a Host or Client clock). The Pdelay_Resp message contains the time at which a Pdelay_Req was received and is itself an event message. The time at which a Pdelay_Resp message is received is returned in a Pdelay_Resp_Follow_Up message.

1588 version 2 introduces transparent clocks of which there are two kinds, peer-to-peer (P2P) and end-to-end (E2E). Transparent clocks measure the transit time of event messages through a bridge and amend a correction field within the message to allow for the transit time. P2P transparent clocks additionally correct for the delay in the receive path of the link using the information gathered from the peer delay frames. With P2P transparent clocks delay_req messages are not used to measure link delay. This simplifies the protocol and makes larger systems more stable.

The GMAC recognizes four different encapsulations for PTP event messages:

1. 1588 version 1 (UDP/IPv4 multicast)
2. 1588 version 2 (UDP/IPv4 multicast)
3. 1588 version 2 (UDP/IPv6 multicast)
4. 1588 version 2 (Ethernet multicast)

**Table 24-5.** Example of Sync Frame in 1588 Version 1 Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | — |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 0800 |
| IP stuff (Octets 14–22) | — |
| UDP (Octet 23) | 11 |
| IP stuff (Octets 24–29) | — |
| IP DA (Octets 30–32) | E00001 |
| IP DA (Octet 33) | 81 or 82 or 83 or 84 |
| Source IP port (Octets 34–35) | — |
| Dest IP port (Octets 36–37) | 013F |
| Other stuff (Octets 38–42) | — |
| Version PTP (Octet 43) | 01 |
| Other stuff (Octets 44–73) | — |
| Control (Octet 74) | 00 |
| Other stuff (Octets 75–168) | — |

**Table 24-6.** Example of Delay Request Frame in 1588 Version 1 Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | — |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 24-6.** Example of Delay Request Frame in 1588 Version 1 Format (continued)

| Frame Segment | Value |
|---|---|
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 0800 |
| IP stuff (Octets 14–22) | — |
| UDP (Octet 23) | 11 |
| IP stuff (Octets 24–29) | — |
| IP DA (Octets 30–32) | E00001 |
| IP DA (Octet 33) | 81 or 82 or 83 or 84 |
| Source IP port (Octets 34–35) | — |
| Dest IP port (Octets 36–37) | 013F |
| Other stuff (Octets 38–42) | — |
| Version PTP (Octet 43) | 01 |
| Other stuff (Octets 44–73) | — |
| Control (Octet 74) | 01 |
| Other stuff (Octets 75–168) | — |

For 1588 version 1 messages, sync and delay request frames are indicated by the GMAC if the frame type field indicates TCP/IP, UDP protocol is indicated, the destination IP address is 224.0.1.129/130/131 or 132, the destination UDP port is 319 and the control field is correct.

The control field is 0x00 for sync frames and 0x01 for delay request frames.

For 1588 version 2 messages, the type of frame is determined by looking at the message type field in the first byte of the PTP frame. Whether a frame is version 1 or version 2 can be determined by looking at the version PTP field in the second byte of both version 1 and version 2 PTP frames.

In version 2 messages sync frames have a message type value of 0x0, delay_req have 0x1, Pdelay_Req have 0x2 and Pdelay_Resp have 0x3.

**Table 24-7.** Example of Sync Frame in 1588 Version 2 (UDP/IPv4) Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | — |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 0800 |
| IP stuff (Octets 14–22) | — |
| UDP (Octet 23) | 11 |
| IP stuff (Octets 24–29) | — |
| IP DA (Octets 30–33) | E0000181 |
| Source IP port (Octets 34–35) | — |
| Dest IP port (Octets 36–37) | 013F |
| Other stuff (Octets 38–41) | — |
| Message type (Octet 42) | 00 |
| Version PTP (Octet 43) | 02 |

**Table 24-8.** Example of Pdelay_Req Frame in 1588 Version 2 (UDP/IPv4) Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | — |
| SA (Octets 6–11) | — |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 24-8.** Example of Pdelay_Req Frame in 1588 Version 2 (UDP/IPv4) Format (continued)

| Frame Segment | Value |
|---|---|
| Type (Octets 12–13) | 0800 |
| IP stuff (Octets 14–22) | — |
| UDP (Octet 23) | 11 |
| IP stuff (Octets 24–29) | — |
| IP DA (Octets 30–33) | E000006B |
| Source IP port (Octets 34–35) | — |
| Dest IP port (Octets 36–37) | 013F |
| Other stuff (Octets 38–41) | — |
| Message type (Octet 42) | 02 |
| Version PTP (Octet 43) | 02 |

**Table 24-9.** Example of Sync Frame in 1588 Version 2 (UDP/IPv6) Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | — |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 86dd |
| IP stuff (Octets 14–19) | — |
| UDP (Octet 20) | 11 |
| IP stuff (Octets 21–37) | — |
| IP DA (Octets 38–53) | FF0X00000000018 |
| Source IP port (Octets 54–55) | — |
| Dest IP port (Octets 56–57) | 013F |
| Other stuff (Octets 58–61) | — |
| Message type (Octet 62) | 00 |
| Other stuff (Octets 63–93) | — |
| Version PTP (Octet 94) | 02 |

**Table 24-10.** Example of Pdelay_Resp Frame in 1588 Version 2 (UDP/IPv6) Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | — |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 86dd |
| IP stuff (Octets 14–19) | — |
| UDP (Octet 20) | 11 |
| IP stuff (Octets 21–37) | — |
| IP DA (Octets 38–53) | FF0200000000006B |
| Source IP port (Octets 54–55) | — |
| Dest IP port (Octets 56–57) | 013F |
| Other stuff (Octets 58–61) | — |
| Message type (Octet 62) | 03 |
| Other stuff (Octets 63–93) | — |
| Version PTP (Octet 94) | 02 |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

For the multicast address 011B19000000 sync and delay request frames are recognized depending on the message type field, 00 for sync and 01 for delay request.

**Table 24-11.** Example of Sync Frame in 1588 Version 2 (Ethernet Multicast) Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | 011B19000000 |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 88F7 |
| Message type (Octet 14) | 00 |
| Version PTP (Octet 15) | 02 |

Pdelay request frames need a special multicast address so they can pass through ports blocked by the spanning tree protocol. For the multicast address 0180C200000E sync, Pdelay_Req and Pdelay_Resp frames are recognized depending on the message type field, 00 for sync, 02 for pdelay request and 03 for pdelay response.

**Table 24-12.** Example of Pdelay_Req Frame in 1588 Version 2 (Ethernet Multicast) Format

| Frame Segment | Value |
|---|---|
| Preamble/SFD | 55555555555555D5 |
| DA (Octets 0–5) | 0180C200000E |
| SA (Octets 6–11) | — |
| Type (Octets 12–13) | 88F7 |
| Message type (Octet 14) | 00 |
| Version PTP (Octet 15) | 02 |

### 24.6.15. Time Stamp Unit

#### Overview

The TSU consists of a timer and registers to capture the time at which PTP event frames cross the message timestamp point. An interrupt is issued when a capture register is updated.

The 1588 time stamp unit (TSU) is implemented as a 94-bit timer.

- The 48 upper bits [93:46] of the timer count seconds and are accessible in the GMAC 1588 Timer Seconds High Register" (TSH) and GMAC 1588 Timer Seconds Low Register (TSL).
- The 30 lower bits [45:16] of the timer count nanoseconds and are accessible in the GMAC 1588 Timer Nanoseconds Register (TN).
- The lowest 16 bits [15:0] of the timer count sub-nanoseconds.

The 46 lower bits roll over when they have counted to 1s. An interrupt is generated when the seconds increment. The timer increments by a programmable period (to approximately 15.2fs resolution) with each MCK period. The timer value can be read, written and adjusted with 1ns resolution (incremented or decremented) through the APB interface.

#### Timer Adjustment

The amount by which the timer increments each clock cycle is controlled by the Timer Increment register (TI). Bits [7:0] are the default increment value in nanoseconds. Additional 16 bits of sub-nanosecond resolution are available using the Timer Increment Sub-Nanoseconds register (TISUBN). If the rest of the register is written with zero, the timer increments by the value in [7:0], plus the value of the TISUBN for each clock cycle.

The TISUBN allows a resolution of approximately 15fs.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Bits [15:8] of the increment register are the alternative increment value in nanoseconds, and bits [23:16] are the number of increments after which the alternative increment value is used. If [23:16] are zero the alternative increment value will never be used.

> Taking the example of 10.2MHz, there are 102 cycles every 10µs or 51 cycles every 5µs. So a timer with a 10.2MHz clock source is constructed by incrementing by 98ns for fifty cycles and then incrementing by 100ns (98ns × 50 + 100ns = 5000ns). This is programmed by writing the value 0x00326462 to the Timer Increment register (TI).

> In a second example, a 49.8 MHz clock source requires 20ns for 248 cycles, followed by an increment of 40ns (20ns × 248 + 40ns = 5000ns). This is programmed by writing the value 0x00F82814 to the TI register.

The Number of Increments bit field in the TI register is 8 bit in size, so frequencies up to 50MHz are supported with 200kHz resolution.

Without the alternative increment field the period of the clock would be limited to an integer number of nanoseconds, resulting in supported clock frequencies of 8, 10, 20, 25, 40, 50, 100, 125, 200 and 250 MHz.

There are eight additional 80-bit registers that capture the time at which PTP event frames are transmitted and received. An interrupt is issued when these registers are updated. The TSU timer count value can be compared to a programmable comparison value. For the comparison, the 48 bits of the seconds value and the upper 22 bits of the nanoseconds value are used. An internal signal (GTSUCOMP) is output from the core to indicate when the TSU timer count value is equal to the comparison value stored in the TSU timer comparison value registers (GMAC.NSC, GMAC.SCL, and GMAC.SCH). An interrupt can also be generated (if enabled) when the TSU timer count value and comparison value are equal, mapped to bit 29 of the interrupt status register.

### 24.6.16. MAC 802.3 Pause Frame Support

**Note:** Refer to the Clause 31, and Annex 31A and 31B of the IEEE standard 802.3 for a full description of MAC 802.3 pause operation.

The following table shows the start of a MAC 802.3 pause frame.

**Table 24-13.** Start of an 802.3 Pause Frame

| Address | | Type (MAC Control Frame) | Pause | |
|---|---|---|---|---|
| Destination | Source | | Opcode | Time |
| 0x0180C2000001 | 6 bytes | 0x8808 | 0x0001 | 2 bytes |

The GMAC supports both hardware controlled pause of the transmitter, upon reception of a pause frame, and hardware generated pause frame transmission.

#### 24.6.16.1. 802.3 Pause Frame Reception

The bit 13 of the Network Configuration register is the pause enable control for reception. If this bit is set, transmission will pause if a non zero pause quantum frame is received.

If a valid pause frame is received, then the Pause Time register is updated with the new frame's pause time, regardless of whether a previous pause frame is active or not. An interrupt (either bit 12 or bit 13 of the Interrupt Status register) is triggered when a pause frame is received, but only if the interrupt has been enabled (bit 12 and bit 13 of the Interrupt Mask register). Pause frames received with non zero quantum are indicated through the interrupt bit 12 of the Interrupt Status register. Pause frames received with zero quantum are indicated on bit 13 of the Interrupt Status register.

Once the Pause Time register is loaded and the frame currently being transmitted has been sent, no new frames are transmitted until the pause time reaches zero. The loading of a new pause time, and hence the pausing of transmission, only occurs when the GMAC is configured for full duplex


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

operation. If the GMAC is configured for half duplex there will be no transmission pause, but the pause frame received interrupt will still be triggered. A valid pause frame is defined as having a destination address that matches either the address stored in Specific Address register '1' or if it matches the reserved address of 0x0180C2000001. It must also have the MAC control frame type ID of 0x8808 and have the pause opcode of 0x0001.

Pause frames that have frame check sequence (FCS) or other errors will be treated as invalid and will be discarded. Valid pause frames received will increment the pause frames received statistic register.

The pause time register decrements every 512 bit times once the transmission has stopped. For test purposes, the retry test bit can be set (bit 12 in the Network Configuration register) which causes the Pause Time register to decrement every GTXCK cycle once transmission has stopped.

The interrupt (bit 13 in the Interrupt Status register) is asserted whenever the Pause Time register decrements to zero (assuming it has been enabled by bit 13 in the Interrupt Mask register). This interrupt is also set when a zero quantum pause frame is received.

#### 24.6.16.2. 802.3 Pause Frame Transmission

Automatic transmission of pause frames is supported through the transmit pause frame bits of the Network Control register. If either bit 11 or bit 12 of the Network Control register is written with logic 1, an 802.3 pause frame will be transmitted, providing full duplex is selected in the Network Configuration register and the transmit block is enabled in the Network Control register.

Pause frame transmission will happen immediately if transmit is inactive or if transmit is active between the current frame and the next frame due to be transmitted.

Transmitted pause frames comprise the following:

- A destination address of 01-80-C2-00-00-01
- A source address taken from Specific Address register 1
- A type ID of 88-08 (MAC control frame)
- A pause opcode of 00-01
- A pause quantum register
- Fill of 00 to take the frame to minimum frame length
- Valid FCS

The pause quantum used in the generated frame will depend on the trigger source for the frame as follows:

- If bit 11 is written with a '1', the pause quantum will be taken from the Transmit Pause Quantum register. The Transmit Pause Quantum register resets to a value of 0xFFFF giving maximum pause quantum as default.
- If bit 12 is written with a '1', the pause quantum will be zero.

After transmission, a pause frame transmitted interrupt will be generated (bit 14 of the Interrupt Status register) and the only statistics register that will be incremented will be the Pause Frames Transmitted register.

Pause frames can also be transmitted by the MAC using normal frame transmission methods.

### 24.6.17. Energy Efficient Ethernet Support

**Features**

- Energy Efficient Ethernet according to IEEE 802.3az
- A system's transmit path can enter a low power mode if there is nothing to transmit.
- A PHY can detect whether its link partner's transmit path is in low power mode, and configure its own receive path to enter low power mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Link remains up during lower power mode and no frames are dropped.
- Asymmetric, one direction can be in low power mode while the other is transmitting normally.
- LPI (Low Power Idle) signaling is used to control entry and exit to and from low power modes.
  **Note:** LPI signaling can only take place if both sides have indicated support for it through auto-negotiation.

**Operation**

- Low power control is done at the MII (reconciliation sublayer).
- As an architectural convenience in writing the 802.3az it is assumed that transmission is deferred by asserting carrier sense - in practice it will not be done this way. This system will know when it has nothing to transmit and only enter low power mode when it is not transmitting.
- LPI should not be requested unless the link has been up for at least one second.
- LPI is signaled on the MII transmit path by asserting 0x01 on txd with tx_en low and tx_er high.
- A PHY on seeing LPI requested on the MII will send the sleep signal before going quiet. After going quiet it will periodically emit refresh signals.
- The sleep, quiet and refresh periods are defined in 802.3az, Table 78-2.
- LPI mode ends by transmitting normal idle for the wake time. There is a default time for this but it can be adjusted in software using the Link Layer Discovery Protocol (LLDP) described in 802.3az, Clause 79.
- LPI is indicated at the receive side when sleep and refresh signaling has been detected.

### 24.6.18. 802.1Qav Support - Credit-based Shaping

A credit-based shaping algorithm is available on the two highest priority queues and is defined in the standard 802.1Qav: Forwarding and Queuing Enhancements for Time-Sensitive Streams. This allows traffic on these queues to be limited and to allow other queues to transmit.

Traffic shaping is enabled via the CBS (Credit Based Shaping) Control register. This enables a counter which stores the amount of transmit 'credit', measured in bytes that a particular queue has. A queue may only transmit if it has non-negative credit. If a queue has data to send, but is held off from doing as another queue is transmitting, then credit will accumulate in the credit counter at the rate defined in the IdleSlope register (CBSISQx) for that queue.

**portTransmitRate** is the transmission rate, in bits per second, that the underlying MAC service that supports transmission through the Port provides. The value of this parameter is determined by the operation of the MAC. IdleSlope is the rate of change of increasing credit when waiting to transmit and must be less than the value of the portTransmitRate.

**IdleSlope** is the rate of change of credit when waiting to transmit and must be less than the value of the portTransmitRate.

The max value of IdleSlope (or sendSlope) is (portTransmitRate / bits_per_MII_Clock).

In case of 100 Mbps, maximum IdleSlope = (100 Mbps / 4) = 0x17D7840.

When this queue is transmitting the credit counter is decremented at the rate of sendSlope which is defined as (portTransmitRate - IdleSlope). A queue can accumulate negative credit when transmitting which will hold off any other transfers from that queue until credit returns to a non-negative value. No transfers are halted when a queue's credit becomes negative; it will accumulate negative credit until the transfer completes.

The highest priority queue always has priority regardless of which queue has the most credit.

### 24.6.19. PHY Interface

Different PHY interfaces are supported by the Ethernet MAC:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- MII
- RMII

The MII interface is provided for 10/100 operation and uses txd[3:0] and rxd[3:0]. The RMII interface is provided for 10/100 operation and uses txd[1:0] and rxd[1:0].

### 24.6.20. 10/100 Operation

The 10/100 Mbps speed bit in the Network Configuration register is used to select between 10 Mbps and 100 Mbps.

### 24.6.21. Jumbo Frames

The jumbo frames enable bit in the Network Configuration register allows the GMAC, in its default configuration, to receive jumbo frames up to 10240 bytes in size. This operation does not form part of the IEEE 802.3 specification and is normally disabled. When jumbo frames are enabled, frames received with a frame size greater than 10240 bytes are discarded.

## 24.7. Programming Interface

### 24.7.1. Initialization

#### 24.7.1.1. Configuration

Initialization of the GMAC configuration (e.g., loop back mode, frequency ratios) must be done while the transmit and receive circuits are disabled. See the description of the Network Control register and Network Configuration register earlier in this document.

To change loop back mode, the following sequence of operations must be followed:

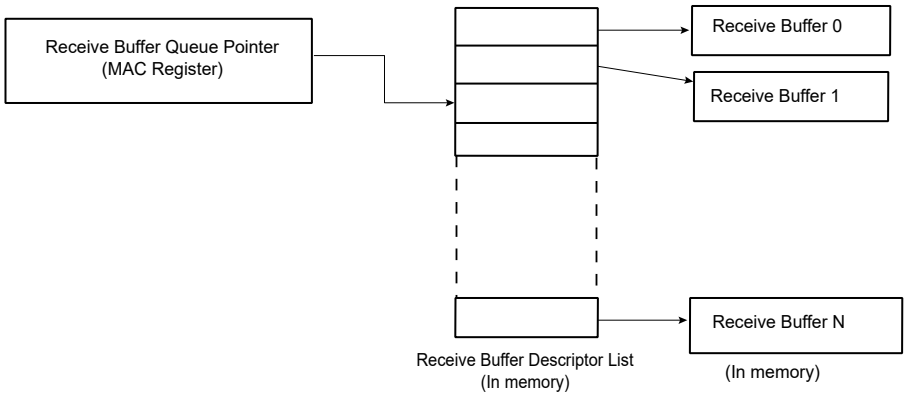
1. Write to Network Control register to disable transmit and receive circuits.
2. Write to Network Control register to change loop back mode.
3. Write to Network Control register to re-enable transmit or receive circuits.
   Note: These writes to the Network Control register cannot be combined in any way.

#### 24.7.1.2. Receive Buffer List

Receive data is written to areas of data (i.e., buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (receive buffer queue) is a sequence of descriptor entries as defined in Table 1-6 “Receive Buffer Descriptor Entry”.

The Receive Buffer Queue Pointer register points to this data structure.

**Figure 24-3.** Receive Buffer List




To create the list of buffers:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

1. Allocate a number (N) of buffers of X bytes in system memory, where X is the DMA buffer length programmed in the DMA Configuration register.
2. Allocate an area 8N bytes for the receive buffer descriptor list in system memory and create N entries in this list. Mark all entries in this list as owned by GMAC, i.e., bit 0 of word 0 set to 0.
3. Mark the last descriptor in the queue with the wrap bit (bit 1 in word 0 set to 1).
4. Write address of receive buffer descriptor list and control information to GMAC register receive buffer queue pointer
5. The receive circuits can then be enabled by writing to the address recognition registers and the Network Control register.

#### 24.7.1.3. Transmit Buffer List

Transmit data is read from areas of data (the buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (Transmit Buffer Queue) is a sequence of descriptor entries as defined in Table 1-7 "Transmit Buffer Descriptor Entry".

The Transmit Buffer Queue Pointer register points to this data structure.

To create this list of buffers:

1. Allocate a number (N) of buffers of between 1 and 2047 bytes of data to be transmitted in system memory. Up to 128 buffers per frame are allowed.
2. Allocate an area 8N bytes for the transmit buffer descriptor list in system memory and create N entries in this list. Mark all entries in this list as owned by GMAC, i.e., bit 31 of word 1 set to 0.
3. Mark the last descriptor in the queue with the wrap bit (bit 30 in word 1 set to 1).
4. Write address of transmit buffer descriptor list and control information to GMAC register transmit buffer queue pointer.
5. The transmit circuits can then be enabled by writing to the Network Control register.

#### 24.7.1.4. Address Matching

The GMAC register pair hash address and the four Specific Address register pairs must be written with the required values. Each register pair comprises of a bottom register and top register, with the bottom register being written first. The address matching is disabled for a particular register pair after the bottom register has been written and re-enabled when the top register is written. Each register pair may be written at any time, regardless of whether the receive circuits are enabled or disabled.

As an example, to set Specific Address register 1 to recognize destination address 21:43:65:87:A9:CB, the following values are written to Specific Address register 1 bottom and Specific Address register 1 top:

- Specific Address register 1 bottom bits 31:0 (0x98): 0x8765_4321.
- Specific Address register 1 top bits 31:0 (0x9C): 0x0000_CBA9.

#### 24.7.1.5. PHY Maintenance

The PHY Maintenance register is implemented as a shift register. Writing to the register starts a shift operation which is signalled as complete when bit two is set in the Network Status register (about 2000 MCK cycles later when bits 18:16 are set to 010 in the Network Configuration register). An interrupt is generated as this bit is set.

During this time, the MSB of the register is output on the MDIO pin and the LSB updated from the MDIO pin with each Management Data Clock (MDC) cycle. This causes the transmission of a PHY management frame on MDIO. See section 22.2.4.5 of the IEEE 802.3 standard.

Reading during the shift operation will return the current contents of the shift register. At the end of the management operation the bits will have shifted back to their original locations. For a read


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

operation the data bits are updated with data read from the PHY. It is important to write the correct values to the register to ensure a valid PHY management frame is produced.

The Management Data Clock (MDC) should not toggle faster than 2.5 MHz (minimum period of 400 ns), as defined by the IEEE 802.3 standard. MDC is generated by dividing down MCK. Three bits in the Network Configuration register determine by how much MCK should be divided to produce MDC.

#### 24.7.1.6. Interrupts

There are 18 interrupt conditions that are detected within the GMAC. The conditions are ORed to make a single interrupt. Depending on the overall system design this may be passed through a further level of interrupt collection (interrupt controller). On receipt of the interrupt signal, the CPU enters the interrupt handler. Refer to the device interrupt controller documentation to identify that it is the GMAC that is generating the interrupt. To ascertain which interrupt, read the Interrupt Status register. Note that in the default configuration this register will clear itself after being read, though this may be configured to be write-one-to-clear if desired.

At reset all interrupts are disabled. To enable an interrupt, write to Interrupt Enable register with the pertinent interrupt bit set to 1. To disable an interrupt, write to Interrupt Disable register with the pertinent interrupt bit set to 1. To check whether an interrupt is enabled or disabled, read Interrupt Mask register. If the bit is set to 1, the interrupt is disabled.

#### 24.7.1.7. Transmitting Frames

The procedure to set up a frame for transmission is the following:

1. Enable transmit in the Network Control register.
2. Allocate an area of system memory for transmit data. This does not have to be contiguous, varying byte lengths can be used if they conclude on byte borders.
3. Set-up the transmit buffer list by writing buffer addresses to word zero of the transmit buffer descriptor entries and control and length to word one.
4. Write data for transmission into the buffers pointed to by the descriptors.
5. Write the address of the first buffer descriptor to transmit buffer descriptor queue pointer.
6. Enable appropriate interrupts.
7. Write to the transmit start bit (TSTART) in the Network Control register.

#### 24.7.1.8. Receiving Frames

When a frame is received and the receive circuits are enabled, the GMAC checks the address and, in the following cases, the frame is written to system memory:

- If it matches one of the four Specific Address registers.
- If it matches one of the four type ID registers.
- If it matches the hash address function.
- If it is a broadcast address (0xFFFFFFFFFFFF) and broadcasts are allowed.
- If the GMAC is configured to “copy all frames”.

The register receive buffer queue pointer points to the next entry in the receive buffer descriptor list and the GMAC uses this as the address in system memory to write the frame to.

Once the frame has been completely and successfully received and written to system memory, the GMAC then updates the receive buffer descriptor entry (see Table 1-6 “Receive Buffer Descriptor Entry”) with the reason for the address match and marks the area as being owned by software. Once this is complete, a receive complete interrupt is set. Software is then responsible for copying the data to the application area and releasing the buffer (by writing the ownership bit back to 0).

If the GMAC is unable to write the data at a rate to match the incoming frame, then a receive overrun interrupt is set. If there is no receive buffer available, i.e., the next buffer is still owned by


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

software, a receive buffer not available interrupt is set. If the frame is not successfully received, a statistics register is incremented and the frame is discarded without informing software.

### 24.7.2. Statistics Registers

Statistics registers are described in the User Interface beginning with Section 1.8.48 "GMAC Octets Transmitted Low Register" and ending with Section 1.8.92 "GMAC UDP Checksum Errors Register".

The statistics register block begins at 0x100 and runs to 0x1B0, and comprises the registers listed below.

| | |
|---|---|
| Octets Transmitted Low Register | Broadcast Frames Received Register |
| Octets Transmitted High Register | Multicast Frames Received Register |
| Frames Transmitted Register | Pause Frames Received Register |
| Broadcast Frames Transmitted Register | 64 Byte Frames Received Register |
| Multicast Frames Transmitted Register | 65 to 127 Byte Frames Received Register |
| Pause Frames Transmitted Register | 128 to 255 Byte Frames Received Register |
| 64 Byte Frames Transmitted Register | 256 to 511 Byte Frames Received Register |
| 65 to 127 Byte Frames Transmitted Register | 512 to 1023 Byte Frames Received Register |
| 128 to 255 Byte Frames Transmitted Register | 1024 to 1518 Byte Frames Received Register |
| 256 to 511 Byte Frames Transmitted Register | 1519 to Maximum Byte Frames Received Register |
| 512 to 1023 Byte Frames Transmitted Register | Undersize Frames Received Register |
| 1024 to 1518 Byte Frames Transmitted Register | Oversize Frames Received Register |
| Greater Than 1518 Byte Frames Transmitted Register | Jabbers Received Register |
| Transmit Underruns Register | Frame Check Sequence Errors Register |
| Single Collision Frames Register | Length Field Frame Errors Register |
| Multiple Collision Frames Register | Receive Symbol Errors Register |
| Excessive Collisions Register | Alignment Errors Register |
| Late Collisions Register | Receive Resource Errors Register |
| Deferred Transmission Frames Register | Receive Overrun Register |
| Carrier Sense Errors Register | IP Header Checksum Errors Register |
| Octets Received Low Register | TCP Checksum Errors Register |
| Octets Received High Register | UDP Checksum Errors Register |
| Frames Received Register | |

These registers reset to zero on a read and stick at all ones when they count to their maximum value. They should be read frequently enough to prevent loss of data.

The receive statistics registers are only incremented when the receive enable bit (RXEN) is set in the Network Control register.

Once a statistics register has been read, it is automatically cleared. When reading the Octets Transmitted and Octets Received registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 24.8. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | NCR | 7:0 | WESTAT | INCSTAT | CLRSTAT | MPE | TXEN | RXEN | LBL | |
| | | 15:8 | SRTSM | | | TXZQPF | TXPF | THALT | TSTART | BP |
| | | 23:16 | | | | | LPI | FNP | TXPBPF | ENPBPR |
| | | 31:24 | | | | | | | | |
| 0x04 | NCFGR | 7:0 | UNIHEN | MTIHEN | NBC | CAF | JFRAME | DNVLAN | FD | SPD |
| | | 15:8 | RXBUFO[1:0] | | PEN | RTY | | | | MAXFS |
| | | 23:16 | DCPF | DBW[1:0] | | | CLK[2:0] | | RFCS | LFERD |
| | | 31:24 | | IRXER | RXBP | IPGSEN | | IRXFCS | EFRHD | RXCOEN |
| 0x08 | NSR | 7:0 | | | | | | IDLE | MDIO | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | UR | 7:0 | | | | | | | | MII |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | DCFGR | 7:0 | ESPA | ESMA | | FBLDO[4:0] | | | | |
| | | 15:8 | | | | | TXCOEN | TXPBMS | RXBMS[1:0] | |
| | | 23:16 | DRBS[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | DDRP |
| 0x14 | TSR | 7:0 | | UND | TXCOMP | TFC | TXGO | RLE | COL | UBR |
| | | 15:8 | | | | | | | | HRESP |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | RBQB | 7:0 | ADDR[5:0] | | | | | | | |
| | | 15:8 | ADDR[13:6] | | | | | | | |
| | | 23:16 | ADDR[21:14] | | | | | | | |
| | | 31:24 | ADDR[29:22] | | | | | | | |
| 0x1C | TBQB | 7:0 | ADDR[5:0] | | | | | | | |
| | | 15:8 | ADDR[13:6] | | | | | | | |
| | | 23:16 | ADDR[21:14] | | | | | | | |
| | | 31:24 | ADDR[29:22] | | | | | | | |
| 0x20 | RSR | 7:0 | | | | | HNO | RXOVR | REC | BNA |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | ISR | 7:0 | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| | | 15:8 | | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| | | 23:16 | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| | | 31:24 | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| 0x28 | IER | 7:0 | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| | | 15:8 | EXINT | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| | | 23:16 | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| | | 31:24 | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| 0x2C | IDR | 7:0 | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| | | 15:8 | EXINT | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| | | 23:16 | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| | | 31:24 | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| 0x30 | IMR | 7:0 | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| | | 15:8 | EXINT | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| | | 23:16 | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| | | 31:24 | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| 0x34 | MAN | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | PHYA[0] | REGA[4:0] | | | | | WTN[1:0] | |
| | | 31:24 | WZO | CLTTO | OP[1:0] | | PHYA[4:1] | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x38 | RPQ | 7:0 | RPQ[7:0] | | | | | | | |
| | | 15:8 | RPQ[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x3C | TPQ | 7:0 | TPQ[7:0] | | | | | | | |
| | | 15:8 | TPQ[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x40 | TPSF | 7:0 | TPB1ADR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | TPB1ADR[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | ENTXP | | | | | | | |
| 0x44 | RPSF | 7:0 | RPB1ADR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | RPB1ADR[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | ENRXP | | | | | | | |
| 0x48 | RJFML | 7:0 | FML[7:0] | | | | | | | |
| | | 15:8 | | | FML[13:8] | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C ... 0x7F | Reserved | | | | | | | | | |
| 0x80 | HRB | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x84 | HRT | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x88 | SAB0 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x8C | SAT0 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x90 | SAB1 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x94 | SAT1 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x98 | SAB2 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x9C | SAT2 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xA0 | SAB3 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xA4 | SAT3 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xA8 | TIDM0 | 7:0 | TID[7:0] | | | | | | | |
| | | 15:8 | TID[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | ENID | | | | | | | |
| 0xAC | TIDM1 | 7:0 | TID[7:0] | | | | | | | |
| | | 15:8 | TID[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | ENID | | | | | | | |
| 0xB0 | TIDM2 | 7:0 | TID[7:0] | | | | | | | |
| | | 15:8 | TID[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | ENID | | | | | | | |
| 0xB4 | TIDM3 | 7:0 | TID[7:0] | | | | | | | |
| | | 15:8 | TID[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | ENID | | | | | | | |
| 0xB8 | WOL | 7:0 | IP[7:0] | | | | | | | |
| | | 15:8 | IP[15:8] | | | | | | | |
| | | 23:16 | | | | | MTI | SA1 | ARP | MAG |
| | | 31:24 | | | | | | | | |
| 0xBC | IPGS | 7:0 | FL[7:0] | | | | | | | |
| | | 15:8 | FL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xC0 | SVLAN | 7:0 | VLAN_TYPE[7:0] | | | | | | | |
| | | 15:8 | VLAN_TYPE[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | ESVLAN | | | | | | | |
| 0xC4 | TPFCP | 7:0 | PEV[7:0] | | | | | | | |
| | | 15:8 | PQ[7:0] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xC8 | SAMB1 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0xCC | SAMT1 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xD0 ... 0xDB | Reserved | | | | | | | | | |
| 0xDC | NSC | 7:0 | NANOSEC[7:0] | | | | | | | |
| | | 15:8 | NANOSEC[15:8] | | | | | | | |
| | | 23:16 | | | | NANOSEC[20:16] | | | | |
| | | 31:24 | | | | | | | | |
| 0xE0 | SCL | 7:0 | SEC[7:0] | | | | | | | |
| | | 15:8 | SEC[15:8] | | | | | | | |
| | | 23:16 | SEC[23:16] | | | | | | | |
| | | 31:24 | SEC[31:24] | | | | | | | |
| 0xE4 | SCH | 7:0 | SEC[7:0] | | | | | | | |
| | | 15:8 | SEC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xE8 | EFTSH | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xEC | EFRSH | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xF0 | PEFTSH | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xF4 | PEFRSH | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xF8 ... 0xFF | Reserved | | | | | | | | | |
| 0x0100 | OTLO | 7:0 | TXO[7:0] | | | | | | | |
| | | 15:8 | TXO[15:8] | | | | | | | |
| | | 23:16 | TXO[23:16] | | | | | | | |
| | | 31:24 | TXO[31:24] | | | | | | | |
| 0x0104 | OTHI | 7:0 | TXO[7:0] | | | | | | | |
| | | 15:8 | TXO[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0108 | FT | 7:0 | FTX[7:0] | | | | | | | |
| | | 15:8 | FTX[15:8] | | | | | | | |
| | | 23:16 | FTX[23:16] | | | | | | | |
| | | 31:24 | FTX[31:24] | | | | | | | |
| 0x010C | BCFT | 7:0 | BFTX[7:0] | | | | | | | |
| | | 15:8 | BFTX[15:8] | | | | | | | |
| | | 23:16 | BFTX[23:16] | | | | | | | |
| | | 31:24 | BFTX[31:24] | | | | | | | |
| 0x0110 | MFT | 7:0 | MFTX[7:0] | | | | | | | |
| | | 15:8 | MFTX[15:8] | | | | | | | |
| | | 23:16 | MFTX[23:16] | | | | | | | |
| | | 31:24 | MFTX[31:24] | | | | | | | |
| 0x0114 | PFT | 7:0 | PFTX[7:0] | | | | | | | |
| | | 15:8 | PFTX[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0118 | BFT64 | 7:0 | NFTX[7:0] | | | | | | | |
| | | 15:8 | NFTX[15:8] | | | | | | | |
| | | 23:16 | NFTX[23:16] | | | | | | | |
| | | 31:24 | NFTX[31:24] | | | | | | | |
| 0x011C | TBFT127 | 7:0 | NFTX[7:0] | | | | | | | |
| | | 15:8 | NFTX[15:8] | | | | | | | |
| | | 23:16 | NFTX[23:16] | | | | | | | |
| | | 31:24 | NFTX[31:24] | | | | | | | |
| 0x0120 | TBFT255 | 7:0 | NFTX[7:0] | | | | | | | |
| | | 15:8 | NFTX[15:8] | | | | | | | |
| | | 23:16 | NFTX[23:16] | | | | | | | |
| | | 31:24 | NFTX[31:24] | | | | | | | |
| 0x0124 | TBFT511 | 7:0 | NFTX[7:0] | | | | | | | |
| | | 15:8 | NFTX[15:8] | | | | | | | |
| | | 23:16 | NFTX[23:16] | | | | | | | |
| | | 31:24 | NFTX[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0128 | TBFT1023 | 7:0 | NFTX[7:0] | | | | | | | |
| | | 15:8 | NFTX[15:8] | | | | | | | |
| | | 23:16 | NFTX[23:16] | | | | | | | |
| | | 31:24 | NFTX[31:24] | | | | | | | |
| 0x012C | TBFT1518 | 7:0 | NFTX[7:0] | | | | | | | |
| | | 15:8 | NFTX[15:8] | | | | | | | |
| | | 23:16 | NFTX[23:16] | | | | | | | |
| | | 31:24 | NFTX[31:24] | | | | | | | |
| 0x0130 | GTBFT1518 | 7:0 | NFTX[7:0] | | | | | | | |
| | | 15:8 | NFTX[15:8] | | | | | | | |
| | | 23:16 | NFTX[23:16] | | | | | | | |
| | | 31:24 | NFTX[31:24] | | | | | | | |
| 0x0134 | TUR | 7:0 | TXUNR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | TXUNR[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0138 | SCF | 7:0 | SCOL[7:0] | | | | | | | |
| | | 15:8 | SCOL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | SCOL[17:16] | |
| | | 31:24 | | | | | | | | |
| 0x013C | MCF | 7:0 | MCOL[7:0] | | | | | | | |
| | | 15:8 | MCOL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | MCOL[17:16] | |
| | | 31:24 | | | | | | | | |
| 0x0140 | EC | 7:0 | XCOL[7:0] | | | | | | | |
| | | 15:8 | | | | | | | XCOL[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0144 | LC | 7:0 | LCOL[7:0] | | | | | | | |
| | | 15:8 | | | | | | | LCOL[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0148 | DTF | 7:0 | DEFT[7:0] | | | | | | | |
| | | 15:8 | DEFT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | DEFT[17:16] | |
| | | 31:24 | | | | | | | | |
| 0x014C | CSE | 7:0 | CSR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | CSR[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0150 | ORLO | 7:0 | RXO[7:0] | | | | | | | |
| | | 15:8 | RXO[15:8] | | | | | | | |
| | | 23:16 | RXO[23:16] | | | | | | | |
| | | 31:24 | RXO[31:24] | | | | | | | |
| 0x0154 | ORHI | 7:0 | RXO[7:0] | | | | | | | |
| | | 15:8 | RXO[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0158 | FR | 7:0 | FRX[7:0] | | | | | | | |
| | | 15:8 | FRX[15:8] | | | | | | | |
| | | 23:16 | FRX[23:16] | | | | | | | |
| | | 31:24 | FRX[31:24] | | | | | | | |
| 0x015C | BCFR | 7:0 | BFRX[7:0] | | | | | | | |
| | | 15:8 | BFRX[15:8] | | | | | | | |
| | | 23:16 | BFRX[23:16] | | | | | | | |
| | | 31:24 | BFRX[31:24] | | | | | | | |
| 0x0160 | MFR | 7:0 | MFRX[7:0] | | | | | | | |
| | | 15:8 | MFRX[15:8] | | | | | | | |
| | | 23:16 | MFRX[23:16] | | | | | | | |
| | | 31:24 | MFRX[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0164 | PFR | 7:0 | PFRX[7:0] | | | | | | | |
| | | 15:8 | PFRX[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0168 | BFR64 | 7:0 | NFRX[7:0] | | | | | | | |
| | | 15:8 | NFRX[15:8] | | | | | | | |
| | | 23:16 | NFRX[23:16] | | | | | | | |
| | | 31:24 | NFRX[31:24] | | | | | | | |
| 0x016C | TBFR127 | 7:0 | NFRX[7:0] | | | | | | | |
| | | 15:8 | NFRX[15:8] | | | | | | | |
| | | 23:16 | NFRX[23:16] | | | | | | | |
| | | 31:24 | NFRX[31:24] | | | | | | | |
| 0x0170 | TBFR255 | 7:0 | NFRX[7:0] | | | | | | | |
| | | 15:8 | NFRX[15:8] | | | | | | | |
| | | 23:16 | NFRX[23:16] | | | | | | | |
| | | 31:24 | NFRX[31:24] | | | | | | | |
| 0x0174 | TBFR511 | 7:0 | NFRX[7:0] | | | | | | | |
| | | 15:8 | NFRX[15:8] | | | | | | | |
| | | 23:16 | NFRX[23:16] | | | | | | | |
| | | 31:24 | NFRX[31:24] | | | | | | | |
| 0x0178 | TBFR1023 | 7:0 | NFRX[7:0] | | | | | | | |
| | | 15:8 | NFRX[15:8] | | | | | | | |
| | | 23:16 | NFRX[23:16] | | | | | | | |
| | | 31:24 | NFRX[31:24] | | | | | | | |
| 0x017C | TBFR1518 | 7:0 | NFRX[7:0] | | | | | | | |
| | | 15:8 | NFRX[15:8] | | | | | | | |
| | | 23:16 | NFRX[23:16] | | | | | | | |
| | | 31:24 | NFRX[31:24] | | | | | | | |
| 0x0180 | TMXBFR | 7:0 | NFRX[7:0] | | | | | | | |
| | | 15:8 | NFRX[15:8] | | | | | | | |
| | | 23:16 | NFRX[23:16] | | | | | | | |
| | | 31:24 | NFRX[31:24] | | | | | | | |
| 0x0184 | UFR | 7:0 | UFRX[7:0] | | | | | | | |
| | | 15:8 | | | | | | | UFRX[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0188 | OFR | 7:0 | OFRX[7:0] | | | | | | | |
| | | 15:8 | | | | | | | OFRX[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x018C | JR | 7:0 | JRX[7:0] | | | | | | | |
| | | 15:8 | | | | | | | JRX[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0190 | FCSE | 7:0 | FCKR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | FCKR[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0194 | LFFE | 7:0 | LFER[7:0] | | | | | | | |
| | | 15:8 | | | | | | | LFER[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0198 | RSE | 7:0 | RXSE[7:0] | | | | | | | |
| | | 15:8 | | | | | | | RXSE[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x019C | AE | 7:0 | AER[7:0] | | | | | | | |
| | | 15:8 | | | | | | | AER[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01A0 | RRE | 7:0 | RXRER[7:0] | | | | | | | |
| | | 15:8 | RXRER[15:8] | | | | | | | |
| | | 23:16 | | | | | | | RXRER[17:16] | |
| | | 31:24 | | | | | | | | |
| 0x01A4 | ROE | 7:0 | RXOVR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | RXOVR[9:8] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01A8 | IHCE | 7:0 | HCKER[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01AC | TCE | 7:0 | TCKER[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01B0 | UCE | 7:0 | UCKER[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01B4 ... 0x01BB | Reserved | | | | | | | | | |
| 0x01BC | TISUBN | 7:0 | LSBTIR[7:0] | | | | | | | |
| | | 15:8 | LSBTIR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01C0 | TSH | 7:0 | TCS[7:0] | | | | | | | |
| | | 15:8 | TCS[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01C4 ... 0x01C7 | Reserved | | | | | | | | | |
| 0x01C8 | TSSSL | 7:0 | VTS[7:0] | | | | | | | |
| | | 15:8 | VTS[15:8] | | | | | | | |
| | | 23:16 | VTS[23:16] | | | | | | | |
| | | 31:24 | VTS[31:24] | | | | | | | |
| 0x01CC | TSSN | 7:0 | VTN[7:0] | | | | | | | |
| | | 15:8 | VTN[15:8] | | | | | | | |
| | | 23:16 | VTN[23:16] | | | | | | | |
| | | 31:24 | | | VTN[29:24] | | | | | |
| 0x01D0 | TSL | 7:0 | TCS[7:0] | | | | | | | |
| | | 15:8 | TCS[15:8] | | | | | | | |
| | | 23:16 | TCS[23:16] | | | | | | | |
| | | 31:24 | TCS[31:24] | | | | | | | |
| 0x01D4 | TN | 7:0 | TNS[7:0] | | | | | | | |
| | | 15:8 | TNS[15:8] | | | | | | | |
| | | 23:16 | TNS[23:16] | | | | | | | |
| | | 31:24 | | | TNS[29:24] | | | | | |
| 0x01D8 | TA | 7:0 | ITDT[7:0] | | | | | | | |
| | | 15:8 | ITDT[15:8] | | | | | | | |
| | | 23:16 | ITDT[23:16] | | | | | | | |
| | | 31:24 | ADJ | | ITDT[29:24] | | | | | |
| 0x01DC | TI | 7:0 | CNS[7:0] | | | | | | | |
| | | 15:8 | ACNS[7:0] | | | | | | | |
| | | 23:16 | NIT[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01E0 | EFTSL | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | RUD[31:24] | | | | | | | |
| 0x01E4 | EFTN | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | | | RUD[29:24] | | | | | |
| 0x01E8 | EFRSL | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | RUD[31:24] | | | | | | | |
| 0x01EC | EFRN | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | | | RUD[29:24] | | | | | |
| 0x01F0 | PEFTSL | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | RUD[31:24] | | | | | | | |
| 0x01F4 | PEFTN | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | | | RUD[29:24] | | | | | |
| 0x01F8 | PEFRSL | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | RUD[31:24] | | | | | | | |
| 0x01FC | PEFRN | 7:0 | RUD[7:0] | | | | | | | |
| | | 15:8 | RUD[15:8] | | | | | | | |
| | | 23:16 | RUD[23:16] | | | | | | | |
| | | 31:24 | | | RUD[29:24] | | | | | |
| 0x0200 ... 0x026F | Reserved | | | | | | | | | |
| 0x0270 | RLPITR | 7:0 | RLPITR[7:0] | | | | | | | |
| | | 15:8 | RLPITR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0274 | RLPITI | 7:0 | RLPITI[7:0] | | | | | | | |
| | | 15:8 | RLPITI[15:8] | | | | | | | |
| | | 23:16 | RLPITI[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0278 | TLPITR | 7:0 | TLPITR[7:0] | | | | | | | |
| | | 15:8 | TLPITR[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x027C | TLPITI | 7:0 | TLPITI[7:0] | | | | | | | |
| | | 15:8 | TLPITI[15:8] | | | | | | | |
| | | 23:16 | TLPITI[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |

## 24.9. Register Description


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.1. GMAC Network Control Register

**Name:** NCR
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LPI | FNP | TXPBPF | ENPBPR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SRTSM | | | TXZQPF | TXPF | THALT | TSTART | BP |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WESTAT | INCSTAT | CLRSTAT | MPE | TXEN | RXEN | LBL | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

**Bit 19 – LPI** Low Power Idle Enable
Writing a '1' to this bit will enable low power idle (LPI) transmission, immediately transmitted on txd and tx_er.

**Bit 18 – FNP** Flush Next Packet
Writing a '1' to this bit will flush the next packet from the external RX DPRAM. Flushing the next packet will only take effect if the DMA is not currently writing a packet already stored in the DPRAM to memory.

**Bit 17 – TXPBPF** Transmit PFC Priority-based Pause Frame
Takes the values stored in the Transmit PFC Pause Register.

**Bit 16 – ENPBPR** Enable PFC Priority-based Pause Reception
Writing a '1' to this bit enables PFC Priority Based Pause Reception capabilities, enabling PFC negotiation and recognition of priority-based pause frames.

| Value | Description |
|---|---|
| 0 | Normal operation |
| 1 | PFC Priority-based Pause frames are recognized |

**Bit 15 – SRTSM** Store Receive Time Stamp to Memory
Writing a '1' to this bit causes the CRC of every received frame to be replaced with the value of the nanoseconds field of the 1588 timer that was captured as the receive frame passed the message time stamp point.

| Value | Description |
|---|---|
| 0 | Normal operation |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | All received frames' CRC is replaced with a time stamp |

**Bit 12 – TXZQPF** Transmit Zero Quantum Pause Frame
Writing a '1' to this bit causes a pause frame with zero quantum to be transmitted.
Writing a '0' to this bit has no effect.

**Bit 11 – TXPF** Transmit Pause Frame
Writing one to this bit causes a pause frame to be transmitted.
Writing a '0' to this bit has no effect.

**Bit 10 – THALT** Transmit Halt
Writing a '1' to this bit halts transmission as soon as any ongoing frame transmission ends.
Writing a '0' to this bit has no effect.

**Bit 9 – TSTART** Start Transmission
Writing a '1' to this bit starts transmission.
Writing a '0' to this bit has no effect.

**Bit 8 – BP** Back Pressure
In 10M or 100M half duplex mode, writing a '1' to this bit forces collisions on all received frames.
Ignored in gigabit half duplex mode.

| Value | Description |
|---|---|
| 0 | Frame collisions are not forced |
| 1 | Frame collisions are forced in 10M and 100M half duplex mode |

**Bit 7 – WESTAT** Write Enable for Statistics Registers
Writing a '1' to this bit makes the statistics registers writable for functional test purposes.

| Value | Description |
|---|---|
| 0 | Statistics Registers are write-protected |
| 1 | Statistics Registers are write-enabled |

**Bit 6 – INCSTAT** Increment Statistics Registers
Writing a '1' to this bit increments all Statistics Registers by one for test purposes.
Writing a '0' to this bit has no effect.
This bit will always read '0'.

**Bit 5 – CLRSTAT** Clear Statistics Registers
Writing a '1' to this bit clears the Statistics Registers.
Writing a '0' to this bit has no effect.
This bit will always read '0'.

**Bit 4 – MPE** Management Port Enable
Writing a '1' to this bit enables the Management Port.
Writing a '0' to this bit disables the Management Port, and forces MDIO to high impedance state and MDC to low impedance.

| Value | Description |
|---|---|
| 0 | Management Port is disabled |
| 1 | Management Port is enabled |

**Bit 3 – TXEN** Transmit Enable
Writing a '1' to this bit enables the GMAC transmitter to send data.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to this bit stops transmission immediately, the transmit pipeline and control registers is cleared, and the Transmit Queue Pointer Register will be set to point to the start of the transmit descriptor list.

| Value | Description |
|---|---|
| 0 | Transmit is disabled |
| 1 | Transmit is enabled |

**Bit 2 – RXEN** Receive Enable

Writing a '1' to this bit enables the GMAC to receive data.
Writing a '0' to this bit stops frame reception immediately, and the receive pipeline is cleared. The Receive Queue Pointer Register is not affected.

| Value | Description |
|---|---|
| 0 | Receive is disabled |
| 1 | Receive is enabled |

**Bit 1 – LBL** Loop Back Local

Writing '1' to this bit connects GTX to GRX, GTXEN to GRXDV, and forces full duplex mode.
GRXCK and GTXCK may malfunction as the GMAC is switched into and out of internal loop back. It is important that receive and transmit circuits have already been disabled when making the switch into and out of internal loop back.

| Value | Description |
|---|---|
| 0 | Loop back local is disabled |
| 1 | Loop back local is enabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.2. GMAC Network Configuration Register

**Name:** NCFGR
**Offset:** 0x004
**Reset:** 0x00080000
**Property:** R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | IRXER | RXBP | IPGSEN | | IRXFCS | EFRHD | RXCOEN |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DCPF | DBW[1:0] | | CLK[2:0] | | | RFCS | LFERD |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RXBUFO[1:0] | | PEN | RTY | | | | MAXFS |
| Access | R/W | R/W | R/W | R/W | | | | R/W |
| Reset | 0 | 0 | 0 | 0 | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | UNIHEN | MTIHEN | NBC | CAF | JFRAME | DNVLAN | FD | SPD |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 30 – IRXER** Ignore IPG GRXER
When this bit is written to '1', the Receive Error signal (GRXER) has no effect on the GMAC operation when Receive Data Valid signal (GRXDV) is low.

**Bit 29 – RXBP** Receive Bad Preamble
When written to '1', frames with non-standard preamble are not rejected.

**Bit 28 – IPGSEN** IP Stretch Enable
Writing a '1' to this bit allows the transmit IPG to increase above 96 bit times, depending on the previous frame length using the IPG Stretch Register.

**Bit 26 – IRXFCS** Ignore RX FCS
For normal operation this bit must be written to zero.
When this bit is written to '1', frames with FCS/CRC errors will not be rejected. FCS error statistics will still be collected for frames with bad FCS, and FCS status will be recorded in the DMA descriptor of the frame.

**Bit 25 – EFRHD** Enable Frames Received in half-duplex
Writing a '1' to this bit enables frames to be received in half-duplex mode while transmitting.

**Bit 24 – RXCOEN** Receive Checksum Offload Enable
Writing a '1' to this bit enables the receive checksum engine, and frames with bad IP, TCP or UDP checksums are discarded.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 23 – DCPF** Disable Copy of Pause Frames
Writing a '1' to this bit prevents valid pause frames from being copied to memory. Pause frames are not copied regardless of the state of the Copy All Frames (CAF) bit, whether a hash match is found or whether a type ID match is identified.
If a destination address match is found, the pause frame will be copied to memory. Note that valid pause frames received will still increment pause statistics and pause the transmission of frames, as required.

**Bits 22:21 – DBW[1:0]** Data Bus Width
The default value for this register is 64 bits.

| Value | Name | Description |
|---|---|---|
| 0 | DBW32 | 32-bit data bus width |
| 1 | DBW64 | 64-bit data bus width |

**Bits 20:18 – CLK[2:0]** MDC Clock Division
These bits must be set according to MCK speed, and determine the number MCK will be divided by to generate Management Data Clock (MDC). For conformance with the 802.3 specification, MDC must not exceed 2.5MHz.
**Note:** MDC is only active during MDIO read and write operations.

| Value | Name | Description |
|---|---|---|
| 0 | MCK_8 | MCK divided by 8 (MCK up to 20MHz) |
| 1 | MCK_16 | MCK divided by 16 (MCK up to 40MHz) |
| 2 | MCK_32 | MCK divided by 32 (MCK up to 80MHz) |
| 3 | MCK_48 | MCK divided by 48 (MCK up to 120MHz) |
| 4 | MCK_64 | MCK divided by 64 (MCK up to 160MHz) |
| 5 | MCK_96 | MCK divided by 96 (MCK up to 240MHz) |

**Bit 17 – RFCS** Remove FCS
Writing this bit to '1' will cause received frames to be written to memory without their frame check sequence (last 4 bytes). The indicated frame length will be reduced by four bytes in this mode.

**Bit 16 – LFERD** Length Field Error Frame Discard
Writing a '1' to this bit discards frames with a measured length shorter than the extracted length field (as indicated by bytes 13 and 14 in a non-VLAN tagged frame). This only applies to frames with a length field less than 0x0600.

**Bits 15:14 – RXBUFO[1:0]** Receive Buffer Offset
These bits determine the number of bytes by which the received data is offset from the start of the receive buffer.

**Bit 13 – PEN** Pause Enable
When written to '1', transmission will pause if a non-zero 802.3 classic pause frame is received and PFC has not been negotiated.

**Bit 12 – RTY** Retry Test
This bit must be written to '0' for normal operation.
When writing a '1' to this bit, the back-off between collisions will always be one slot time. This setting helps testing the too many retries condition. This setting is also useful for pause frame tests by reducing the pause counter's decrement time from "512 bit times" to "every GRXCK cycle".

**Bit 8 – MAXFS** 1536 Maximum Frame Size
Writing a '1' to this bit increases the maximum accepted frame size to 1536 bytes in length. When written to '0', any frame above 1518 bytes in length is rejected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 7 – UNIHEN** Unicast Hash Enable
When writing a '1' to this bit, unicast frames will be accepted when the 6-bit hash function of the destination address points to a bit that is set in the Hash Register.
Writing a '0' to this bit disables unicast hashing.

**Bit 6 – MTIHEN** Multicast Hash Enable
When writing a '1' to this bit, multicast frames will be accepted when the 6-bit hash function of the destination address points to a bit that is set in the Hash Register.
Writing a '0' to this bit disables multicast hashing.

**Bit 5 – NBC** No Broadcast
Writing a '1' to this bit will reject frames addressed to the broadcast address 0xFFFFFFFFFFFF (all '1').
Writing a '0' to this bit allows broadcasting to 0xFFFFFFFFFFFF.

**Bit 4 – CAF** Copy All Frames
When writing a '1' to this bit, all valid frames will be accepted.

**Bit 3 – JFRAME** Jumbo Frame Size
Writing a '1' to this bit enables jumbo frames of up to 10240 bytes to be accepted. The default length is 10240 bytes.

**Bit 2 – DNVLAN** Discard Non-VLAN Frames
Writing a '1' to this bit allows only VLAN-tagged frames to pass to the address matching logic.
Writing a '0' to this bit allows both VLAN_tagged and untagged frames to pass to the address matching logic.

**Bit 1 – FD** Full Duplex
Writing a '1' enables full duplex operation, so the transmit block ignores the state of collision and carrier sense and allows receive while transmitting.
Writing a '0' disables full duplex operation.

**Bit 0 – SPD** Speed
Writing a '1' selects 100Mbps operation.
Writing a '0' to this bit selects 10Mbps operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.3. GMAC Network Status Register

**Name:** NSR
**Offset:** 0x008
**Reset:** 0x00000004
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | IDLE | MDIO | |
| Access | | | | | | R | R | |
| Reset | | | | | | 1 | 0 | |

**Bit 2 – IDLE** PHY Management Logic Idle
The PHY management logic is idle (i.e., has completed).

**Bit 1 – MDIO** MDIO Input Status
Returns status of the MDIO pin.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.4. GMAC User Register

**Name:** UR
**Offset:** 0x00C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | MII |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – MII** Reduced MII Mode

| Value | Description |
|---|---|
| 0 | RMII mode is selected |
| 1 | MII mode is selected |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 24.9.5. GMAC DMA Configuration Register

**Name:** DCFGR
**Offset:** 0x010
**Reset:** 0x00020004
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DDRP |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DRBS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | TXCOEN | TXPBMS | RXBMS[1:0] | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ESPA | ESMA | | FBLDO[4:0] | | | | |
| Access | R/W | R/W | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | 0 | 0 | 1 | 0 | 0 |

**Bit 24 – DDRP** DMA Discard Receive Packets
A write to this bit is ignored if the DMA is not configured in the packet buffer full store and forward mode.

| Value | Description |
|---|---|
| 0 | Received packets are stored in the SRAM based packet buffer until next AHB buffer resource becomes available. |
| 1 | Receive packets from the receiver packet buffer memory are automatically discarded when no AHB resource is available. |

**Bits 23:16 – DRBS[7:0]** DMA Receive Buffer Size
These bits defined by these bits determines the size of buffer to use in main AHB system memory when writing received data.
The value is defined in multiples of 64 bytes. For example:

- 0x02: 128 bytes
- 0x18: 1536 bytes (1 × max length frame/buffer)
- 0xA0: 10240 bytes (1 × 10K jumbo frame/buffer)

**WARNING** Do not write 0x00 to this bit field.

| Value | Description |
|---|---|
| 0x00 | Reserved |
| 0x01-0xFF | 1..255 x 64 byte buffer |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 11 – TXCOEN** Transmitter Checksum Generation Offload Enable
Transmitter IP, TCP and UDP checksum generation offload enable.

| Value | Description |
|---|---|
| 0 | Frame data is unaffected. |
| 1 | The transmitter checksum generation engine calculates and substitutes checksums for transmit frames. |

**Bit 10 – TXPBMS** Transmitter Packet Buffer Memory Size Select
When written to zero, the amount of memory used for the transmit packet buffer is reduced by 50%. This reduces the amount of memory used by the GMAC.
It is important to write this bit to '1' if the full configured physical memory is available. The value in parentheses represents the size that would result for the default maximum configured memory size of 4KBytes.

| Value | Description |
|---|---|
| 0 | Top address bits not used. (2KByte used.) |
| 1 | Full configured addressable space (4KBytes) used. |

**Bits 9:8 – RXBMS[1:0]** Receiver Packet Buffer Memory Size Select
The default receive packet buffer size is FULL=RECEIVE_BUFFER_SIZE = 4KB Kbytes. The table below shows how to configure this memory to FULL, HALF, QUARTER or EIGHTH of the default size.

| Value | Name | Description |
|---|---|---|
| 0 | EIGHTH | RECEIVE_BUFFER_SIZE = 4KB/8 Kbyte Memory Size |
| 1 | QUARTER | RECEIVE_BUFFER_SIZE = 4KB/4 Kbytes Memory Size |
| 2 | HALF | RECEIVE_BUFFER_SIZE = 4KB/2 Kbytes Memory Size |
| 3 | FULL | RECEIVE_BUFFER_SIZE = 4KB Kbytes Memory Size |

**Bit 7 – ESPA** Endian Swap Mode Enable for Packet Data Accesses

| Value | Description |
|---|---|
| 0 | Little endian mode for AHB transfers selected. |
| 1 | Big endian mode for AHB transfers selected. |

**Bit 6 – ESMA** Endian Swap Mode Enable for Management Descriptor Accesses

| Value | Description |
|---|---|
| 0 | Little endian mode for AHB transfers selected. |
| 1 | Big endian mode for AHB transfers selected. |

**Bits 4:0 – FBLDO[4:0]** Fixed Burst Length for DMA Data Operations
Selects the burst length to attempt to use on the AHB when transferring frame data. Not used for DMA management operations and only used where space and data size allow. Otherwise SINGLE type AHB transfers are used.
One-hot priority encoding enforced automatically on register writes as follows. 'x' represents don't care.

| Value | Name | Description |
|---|---|---|
| 0 | - | Reserved |
| 1 | SINGLE | 00001: Always use SINGLE AHB bursts |
| 2 | - | Reserved |
| 4 | INCR4 | 001xx: Attempt to use INCR4 AHB bursts (Default) |
| 8 | INCR8 | 01xxx: Attempt to use INCR8 AHB bursts |
| 16 | INCR16 | 1xxxx: Attempt to use INCR16 AHB bursts |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.6. GMAC Transmit Status Register

**Name:** TSR
**Offset:** 0x014
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | HRESP |
| Reset | | | | | | | | R/W |
| | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | UND | TXCOMP | TFC | TXGO | RLE | COL | UBR |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 8 – HRESP** HRESP Not OK
Set when the DMA block sees HRESP not OK.
This bit is cleared by writing a '1' to it.

**Bit 6 – UND** Transmit Underrun
This bit is set if the transmitter was forced to terminate the transmission of a frame due to further data being unavailable.
This bit is also set if a transmitter status write back has not completed when another status write back is attempted.
When using the DMA interface configured for internal FIFO mode, this bit is also set when the transmit DMA has written the SOP data into the FIFO and either the AHB bus was not granted in time for further data, or an AHB not OK response was returned, or a used bit was read.
This bit is cleared by writing a '1' to it.

**Bit 5 – TXCOMP** Transmit Complete
Set when a frame has been transmitted.
This bit is cleared by writing a '1' to it.

**Bit 4 – TFC** Transmit Frame Corruption Due to AHB Error
This bit is set when an error occurs during reading transmit frame from the AHB. Error causes include HRESP errors and buffers exhausted mid frame. (If the buffers run out during transmission of a frame then transmission stops, FCS shall be bad and GTXER asserted).
In DMA packet buffer mode, this bit is also set if a single frame is too large for the configured packet buffer memory size.
This bit is cleared by writing a '1' to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 3 – TXGO** Transmit Go
This bit is '1' when transmit is active. When using the DMA interface this bit represents the TXGO variable as specified in the transmit buffer description.

**Bit 2 – RLE** Retry Limit Exceeded
This bit is cleared by writing a '1' to it.

**Bit 1 – COL** Collision Occurred
When operating in 10/100Mbps mode, this bit is set by the assertion of either a collision or a late collision.
This bit is cleared by writing a '1' to it.

**Bit 0 – UBR** Used Bit Read
This bit is set when a transmit buffer descriptor is read with its used bit set.
This bit is cleared by writing a '1' to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.7. GMAC Receive Buffer Queue Base Address Register

**Name:** RBQB
**Offset:** 0x018
**Reset:** 0x00000000
**Property:** Read/Write

This register holds the start address of the receive buffer queue (receive buffers descriptor list). The receive buffer queue base address must be initialized before receive is enabled through bit 2 of the Network Control Register. Once reception is enabled, any write to the Receive Buffer Queue Base Address Register is ignored. Reading this register returns the location of the descriptor currently being accessed. This value increments as buffers are used. Software should not use this register for determining where to remove received frames from the queue as it constantly changes as new frames are received. Software should instead work its way through the buffer descriptor queue checking the “used” bits.

In terms of AMBA AHB operation, the descriptors are read from memory using a single 32-bit AHB access. The descriptors should be aligned at 32-bit boundaries and the descriptors are written to using two individual non sequential accesses.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[29:22] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[21:14] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[13:6] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bits 31:2 – ADDR[29:0]** Receive Buffer Queue Base Address
Written with the address of the start of the receive queue.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.8. GMAC Transmit Buffer Queue Base Address Register

**Name:** TBQB
**Offset:** 0x01C
**Reset:** 0x00000000
**Property:** -

This register holds the start address of the transmit buffer queue (transmit buffers descriptor list). The Transmit Buffer Queue Base Address Register must be initialized before transmit is started through bit 9 of the Network Control Register. Once transmission has started, any write to the Transmit Buffer Queue Base Address Register is illegal and therefore ignored.

Note that due to clock boundary synchronization, it takes a maximum of four MCK cycles from the writing of the transmit start bit before the transmitter is active. Writing to the Transmit Buffer Queue Base Address Register during this time may produce unpredictable results.

Reading this register returns the location of the descriptor currently being accessed. Since the DMA handles two frames at once, this may not necessarily be pointing to the current frame being transmitted.

In terms of AMBA AHB operation, the descriptors are written to memory using a single 32-bit AHB access. The descriptors should be aligned at 32-bit boundaries and the descriptors are read from memory using two individual non sequential accesses.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[29:22] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[21:14] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[13:6] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bits 31:2 – ADDR[29:0]** Transmit Buffer Queue Base Address
Written with the address of the start of the transmit queue.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.9. GMAC Receive Status Register

**Name:** RSR
**Offset:** 0x020
**Reset:** 0x00000000
**Property:** -

This register, when read, provides receive status details. Once read, individual bits may be cleared by writing a '1' to them. It is not possible to set a bit to '1' by writing to this register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | HNO | RXOVR | REC | BNA |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 3 – HNO** HRESP Not OK
This bit is set when the DMA block sees HRESP not OK.
This bit is cleared by writing a '1' to it.

**Bit 2 – RXOVR** Receive Overrun
This bit is set if the receive status was not taken at the end of the frame. The buffer will be recovered if an overrun occurs.
This bit is cleared by writing a '1' to it.

**Bit 1 – REC** Frame Received
This bit is set to when one or more frames have been received and placed in memory.
This bit is cleared by writing a '1' to it.

**Bit 0 – BNA** Buffer Not Available
When this bit is set, an attempt was made to get a new buffer and the pointer indicated that it was owned by the processor. The DMA will re-read the pointer each time an end of frame is received until a valid pointer is found. This bit is set following each descriptor read attempt that fails, even if consecutive pointers are unsuccessful and software has in the mean time cleared the status flag.
This bit is cleared by writing a '1' to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.10. GMAC Interrupt Status Register

**Name:** ISR
**Offset:** 0x024
**Reset:** 0x00000000
**Property:** -

This register indicates the source of the interrupt. An interrupt source must be enabled in the mask register first so the corresponding bits of this register will be set and the GMAC interrupt signal will be asserted in the system.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| Access | R | R | R | R | R | R | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| Access | | R | R | R | R | R | | |
| Reset | | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 29 – TSUCMP** TSU Timer Comparison
Indicates TSU times count and comparison value are equal.

**Bit 28 – WOL** Wake On LAN
WOL interrupt. Indicates a WOL message has been received.

**Bit 27 – RXLPISBC** Receive LPI indication Status Bit Change
Receive LPI indication status bit change.
Cleared on read.

**Bit 26 – SRI** TSU Seconds Register Increment
Indicates the register has incremented.
Cleared on read.

**Bit 25 – PDRSFT** PDelay Response Frame Transmitted
Indicates a PTP pdelay_resp frame has been transmitted.
Cleared on read.

**Bit 24 – PDRQFT** PDelay Request Frame Transmitted
Indicates a PTP pdelay_req frame has been transmitted.
Cleared on read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 23 – PDRSFR** PDelay Response Frame Received
Indicates a PTP pdelay_resp frame has been received.
Cleared on read.

**Bit 22 – PDRQFR** PDelay Request Frame Received
Indicates a PTP pdelay_req frame has been received.
Cleared on read.

**Bit 21 – SFT** PTP Sync Frame Transmitted
Indicates a PTP sync frame has been transmitted.
Cleared on read.

**Bit 20 – DRQFT** PTP Delay Request Frame Transmitted
Indicates a PTP delay_req frame has been transmitted.
Cleared on read.

**Bit 19 – SFR** PTP Sync Frame Received
Indicates a PTP sync frame has been received.
Cleared on read.

**Bit 18 – DRQFR** PTP Delay Request Frame Received
Indicates a PTP delay_req frame has been received.
Cleared on read.

**Bit 14 – PFTR** Pause Frame Transmitted
Indicates a pause frame has been successfully transmitted after being initiated from the Network Control Register.
Cleared on read.

**Bit 13 – PTZ** Pause Time Zero
Set when either the Pause Time Register at address 0x38 decrements to zero, or when a valid pause frame is received with a zero pause quantum field.
Cleared on read.

**Bit 12 – PFNZ** Pause Frame with Non-zero Pause Quantum Received
Indicates a valid pause has been received that has a non-zero pause quantum field.
Cleared on read.

**Bit 11 – HRESP** HRESP Not OK
Set when the DMA block sees HRESP not OK.
Cleared on read.

**Bit 10 – ROVR** Receive Overrun
Set when the receive overrun status bit is set.
Cleared on read.

**Bit 7 – TCOMP** Transmit Complete
Set when a frame has been transmitted.
Cleared on read.

**Bit 6 – TFC** Transmit Frame Corruption Due to AHB Error
Transmit frame corruption due to AHB error. Set if an error occurs during reading a transmit frame from the AHB, including HRESP errors and buffers exhausted mid frame.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 5 – RLEX** Retry Limit Exceeded
Retry Limit Exceeded Transmit error.
Cleared on read.

**Bit 4 – TUR** Transmit Underrun
This interrupt is set if the transmitter was forced to terminate an ongoing frame transmission due to further data being unavailable.
This interrupt is also set if a transmitter status write back has not completed when another status write back is attempted.
This interrupt is also set when the transmit DMA has written the SOP data into the FIFO and either the AHB bus was not granted in time for further data, or because an AHB not OK response was returned, or because the used bit was read.

**Bit 3 – TXUBR** TX Used Bit Read
Set when a transmit buffer descriptor is read with its used bit set.
Cleared on read.

**Bit 2 – RXUBR** RX Used Bit Read
Set when a receive buffer descriptor is read with its used bit set.
Cleared on read.

**Bit 1 – RCOMP** Receive Complete
A frame has been stored in memory.
Cleared on read.

**Bit 0 – MFS** Management Frame Sent
The PHY Maintenance Register has completed its operation.
Cleared on read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.11. GMAC Interrupt Enable Register

**Name:** IER
**Offset:** 0x028
**Reset:** –
**Property:** Write-only

This register is write-only and will always return zero.

The following values are valid for all listed bit names of this register:

0: No effect.

1: Enables the corresponding interrupt.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| Access | | | W | W | W | W | W | W |
| Reset | | | – | – | – | – | – | – |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| Access | W | W | W | W | W | W | | |
| Reset | – | – | – | – | – | – | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXINT | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| Access | W | W | W | W | W | W | | |
| Reset | – | – | – | – | – | – | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| Access | W | W | W | W | W | W | W | W |
| Reset | – | – | – | – | – | – | – | – |

**Bit 29 – TSUCMP** TSU Timer Comparison

**Bit 28 – WOL** Wake On LAN

**Bit 27 – RXLPISBC** Enable RX LPI Indication

**Bit 26 – SRI** TSU Seconds Register Increment

**Bit 25 – PDRSFT** PDelay Response Frame Transmitted

**Bit 24 – PDRQFT** PDelay Request Frame Transmitted

**Bit 23 – PDRSFR** PDelay Response Frame Received

**Bit 22 – PDRQFR** PDelay Request Frame Received

**Bit 21 – SFT** PTP Sync Frame Transmitted

**Bit 20 – DRQFT** PTP Delay Request Frame Transmitted

**Bit 19 – SFR** PTP Sync Frame Received


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 18 – DRQFR** PTP Delay Request Frame Received

**Bit 15 – EXINT** External Interrupt

**Bit 14 – PFTR** Pause Frame Transmitted

**Bit 13 – PTZ** Pause Time Zero

**Bit 12 – PFNZ** Pause Frame with Non-zero Pause Quantum Received

**Bit 11 – HRESP** HRESP Not OK

**Bit 10 – ROVR** Receive Overrun

**Bit 7 – TCOMP** Transmit Complete

**Bit 6 – TFC** Transmit Frame Corruption Due to AHB Error

**Bit 5 – RLEX** Retry Limit Exceeded or Late Collision

**Bit 4 – TUR** Transmit Underrun

**Bit 3 – TXUBR** TX Used Bit Read

**Bit 2 – RXUBR** RX Used Bit Read

**Bit 1 – RCOMP** Receive Complete

**Bit 0 – MFS** Management Frame Sent


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 24.9.12. GMAC Interrupt Disable Register

**Name:** IDR
**Offset:** 0x02C
**Reset:** –
**Property:** Write-only

This register is write-only and will always return zero.

The following values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| Access | | | W | W | W | W | W | W |
| Reset | | | – | – | – | – | – | – |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| Access | W | W | W | W | W | W | | |
| Reset | – | – | – | – | – | – | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXINT | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| Access | W | W | W | W | W | W | | |
| Reset | – | – | – | – | – | – | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| Access | W | W | W | W | W | W | W | W |
| Reset | – | – | – | – | – | – | – | – |

**Bit 29 – TSUCMP** TSU Timer Comparison

**Bit 28 – WOL** Wake On LAN

**Bit 27 – RXLPISBC** Enable RX LPI Indication

**Bit 26 – SRI** TSU Seconds Register Increment

**Bit 25 – PDRSFT** PDelay Response Frame Transmitted

**Bit 24 – PDRQFT** PDelay Request Frame Transmitted

**Bit 23 – PDRSFR** PDelay Response Frame Received

**Bit 22 – PDRQFR** PDelay Request Frame Received

**Bit 21 – SFT** PTP Sync Frame Transmitted

**Bit 20 – DRQFT** PTP Delay Request Frame Transmitted

**Bit 19 – SFR** PTP Sync Frame Received


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 18 – DRQFR** PTP Delay Request Frame Received

**Bit 15 – EXINT** External Interrupt

**Bit 14 – PFTR** Pause Frame Transmitted

**Bit 13 – PTZ** Pause Time Zero

**Bit 12 – PFNZ** Pause Frame with Non-zero Pause Quantum Received

**Bit 11 – HRESP** HRESP Not OK

**Bit 10 – ROVR** Receive Overrun

**Bit 7 – TCOMP** Transmit Complete

**Bit 6 – TFC** Transmit Frame Corruption Due to AHB Error

**Bit 5 – RLEX** Retry Limit Exceeded or Late Collision

**Bit 4 – TUR** Transmit Underrun

**Bit 3 – TXUBR** TX Used Bit Read

**Bit 2 – RXUBR** RX Used Bit Read

**Bit 1 – RCOMP** Receive Complete

**Bit 0 – MFS** Management Frame Sent


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 24.9.13. GMAC Interrupt Mask Register

**Name:** IMR
**Offset:** 0x030
**Reset:** 0x3FFFFFFF
**Property:** -

This register is a read-only register indicating which interrupts are masked. All bits are set at Reset and can be reset individually by writing to the Interrupt Enable Register (IER), or set individually by writing to the Interrupt Disable Register (IDR).

For test purposes there is a write-only function to this register that allows the bits in the Interrupt Status Register to be set or cleared, regardless of the state of the mask register. A write to this register directly affects the state of the corresponding bit in the Interrupt Status Register, causing an interrupt to be generated if a 1 is written.

The following values are valid for all listed bit names of this register when read:

0: The corresponding interrupt is enabled.

1: The corresponding interrupt is not enabled.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | TSUCMP | WOL | RXLPISBC | SRI | PDRSFT | PDRQFT |
| Access | | | R | R | R | R | R | R |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PDRSFR | PDRQFR | SFT | DRQFT | SFR | DRQFR | | |
| Access | R | R | R | R | R | R | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EXINT | PFTR | PTZ | PFNZ | HRESP | ROVR | | |
| Access | R | R | R | R | R | R | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCOMP | TFC | RLEX | TUR | TXUBR | RXUBR | RCOMP | MFS |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bit 29 – TSUCMP** TSU Timer Comparison
Indicates TSU times count and comparison value are equal.

**Bit 28 – WOL** Wake On LAN
WOL interrupt. Indicates a WOL message has been received.

**Bit 27 – RXLPISBC** Enable RX LPI Indication

**Bit 26 – SRI** TSU Seconds Register Increment
Indicates the register has incremented.
Cleared on read.

**Bit 25 – PDRSFT** PDelay Response Frame Transmitted


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 24 – PDRQFT** PDelay Request Frame Transmitted

**Bit 23 – PDRSFR** PDelay Response Frame Received

**Bit 22 – PDRQFR** PDelay Request Frame Received

**Bit 21 – SFT** PTP Sync Frame Transmitted

**Bit 20 – DRQFT** PTP Delay Request Frame Transmitted

**Bit 19 – SFR** PTP Sync Frame Received

**Bit 18 – DRQFR** PTP Delay Request Frame Received

**Bit 15 – EXINT** External Interrupt

**Bit 14 – PFTR** Pause Frame Transmitted

**Bit 13 – PTZ** Pause Time Zero

**Bit 12 – PFNZ** Pause Frame with Non-zero Pause Quantum Received

**Bit 11 – HRESP** HRESP Not OK

**Bit 10 – ROVR** Receive Overrun

**Bit 7 – TCOMP** Transmit Complete

**Bit 6 – TFC** Transmit Frame Corruption Due to AHB Error

**Bit 5 – RLEX** Retry Limit Exceeded

**Bit 4 – TUR** Transmit Underrun

**Bit 3 – TXUBR** TX Used Bit Read

**Bit 2 – RXUBR** RX Used Bit Read

**Bit 1 – RCOMP** Receive Complete

**Bit 0 – MFS** Management Frame Sent


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 24.9.14. GMAC PHY Maintenance Register

**Name:** MAN
**Offset:** 0x034
**Reset:** 0x00000000
**Property:** Read/Write

This register is a shift register. Writing to it starts a shift operation which is signaled completed when bit 2 is set in the Network Status Register (NSR). It takes about 2000 MCK cycles to complete, when MDC is set for MCK divide by 32 in the Network Configuration Register. An interrupt is generated upon completion.

During this time, the MSB of the register is output on the MDIO pin and the LSB updated from the MDIO pin with each MDC cycle. This causes transmission of a PHY management frame on MDIO. Refer also to section 22.2.4.5 of the IEEE 802.3 standard.

Reading during the shift operation returns the current contents of the shift register. At the end of management operation, the bits will have shifted back to their original locations. For a read operation, the data bits are updated with data read from the PHY. It is important to write the correct values to the register to ensure a valid PHY management frame is produced.

The MDIO interface can read IEEE 802.3 clause 45 PHYs, as well as clause 22 PHYs. To read clause 45 PHYs, bit 30 should be written with a '0' rather than a '1'. To write clause 45 PHYs, bits 31:28 should be written as 0x1:

| PHY | Access | Bit Value | | | |
|---|---|---|---|---|---|
| | | WZO | CLTTO | OP[1] | OP[0] |
| Clause 22 | Read | 0 | 1 | 1 | 0 |
| | Write | 0 | 1 | 0 | 1 |
| Clause 45 | Read | 0 | 0 | 1 | 1 |
| | Write | 0 | 0 | 0 | 1 |
| | Read + Address | 0 | 0 | 1 | 0 |

For a description of MDC generation, see also the 'GMAC Network Configuration Register' (NCR) description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | WZO | CLTTO | OP[1:0] | | PHYA[4:1] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PHYA[0] | REGA[4:0] | | | | | WTN[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – WZO** Write ZERO
Must be written to '0'.

| Value | Description |
|---|---|
| 0 | Mandatory |
| 1 | Reserved |

**Bit 30 – CLTTO** Clause 22 Operation

| Value | Description |
|---|---|
| 0 | Clause 45 operation |
| 1 | Clause 22 operation |

**Bits 29:28 – OP[1:0]** Operation

| Value | Description |
|---|---|
| 01 | Write |
| 10 | Read |
| Other | Reseved |

**Bits 27:23 – PHYA[4:0]** PHY Address

**Bits 22:18 – REGA[4:0]** Register Address
Specifies the register in the PHY to access.

**Bits 17:16 – WTN[1:0]** Write Ten
Must be written to '10'.

| Value | Description |
|---|---|
| 10 | Mandatory |
| Other | Reserved |

**Bits 15:0 – DATA[15:0]** PHY Data
For a write operation, this field is written with the data to be written to the PHY.
After a read operation, this field contains the data read from the PHY.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.15. GMAC Receive Pause Quantum Register

**Name:** RPQ
**Offset:** 0x038
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RPQ[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RPQ[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RPQ[15:0]** Received Pause Quantum
Stores the current value of the Receive Pause Quantum Register which is decremented every 512 bit times.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.16. GMAC Transmit Pause Quantum Register

**Name:** TPQ
**Offset:** 0x03C
**Reset:** 0x0000FFFF
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TPQ[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TPQ[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 15:0 – TPQ[15:0]** Transmit Pause Quantum
Written with the pause quantum value for pause frame transmission.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.17. GMAC TX Partial Store and Forward Register

**Name:** TPSF
**Offset:** 0x040
**Reset:** 0x00000FFF
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ENTXP | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | TPB1ADR[9:8] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TPB1ADR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bit 31 – ENTXP** Enable TX Partial Store and Forward Operation

**Bits 9:0 – TPB1ADR[9:0]** Transmit Partial Store and Forward Address Watermark value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.18. GMAC RX Partial Store and Forward Register

**Name:** RPSF
**Offset:** 0x044
**Reset:** 0x00000FFF
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ENRXP | | | | | | | |
| Access | R | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RPB1ADR[9:8] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RPB1ADR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bit 31 – ENRXP** Enable RX Partial Store and Forward Operation

**Bits 9:0 – RPB1ADR[9:0]** Receive Partial Store and Forward Address Watermark value. Reset = 1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.19. GMAC RX Jumbo Frame Max Length Register

**Name:** RJFML
**Offset:** 0x048
**Reset:** 0x00003FFF
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | FML[13:8] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FML[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 13:0 – FML[13:0]** Frame Max Length
Rx jumbo frame maximum length.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.20. GMAC Hash Register Bottom

**Name:** HRB
**Offset:** 0x080
**Reset:** 0x00000000
**Property:** Read/Write

The unicast hash enable (UNIHEN) and the multicast hash enable (MITIHEN) bits in the Network Configuration Register (NCFGR) enable the reception of hash matched frames.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADDR[31:0]** Hash Address
The first 32 bits of the Hash Address Register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.21. GMAC Hash Register Top

**Name:** HRT
**Offset:** 0x084
**Reset:** 0x00000000
**Property:** Read/Write

The Unicast Hash Enable (UNIHEN) and the Multicast Hash Enable (MITIHEN) bits in the Network Configuration Register (NCFGR) enable the reception of hash matched frames.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADDR[31:0]** Hash Address
Bits 63 to 32 of the Hash Address Register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.22. GMAC Specific Address n Bottom Register

**Name:** SAB
**Offset:** 0x88 + n*0x08 [n=0..3]
**Reset:** 0x00000000
**Property:** -

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADDR[31:0]** Specific Address n
Least significant 32 bits of the destination address, that is, bits 31:0. Bit zero indicates whether the address is multicast or unicast and corresponds to the least significant bit of the first byte received.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.23. GMAC Specific Address n Top Register

**Name:** SAT
**Offset:** 0x8C + n*0x08 [n=0..3]
**Reset:** 0x00000000
**Property:** -

The addresses stored in the Specific Address Registers are deactivated at reset or when their corresponding Specific Address Register Bottom is written. They are activated when Specific Address Register Top is written.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – ADDR[15:0]** Specific Address n
The most significant bits of the destination address, that is, bits 47:32.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.24. GMAC Type ID Match n Register

**Name:** TIDM
**Offset:** 0xA8 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ENID | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TID[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TID[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – ENID** Enable Copying of TID n Matched Frames

| Value | Description |
|---|---|
| 0 | TID n is not part of the comparison match. |
| 1 | TID n is processed for the comparison match. |

**Bits 15:0 – TID[15:0]** Type ID Match n
For use in comparisons with received frames type ID/length frames.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.25. GMAC Wake on LAN Register

**Name:** WOL
**Offset:** 0x0B8
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | MTI | SA1 | ARP | MAG |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | IP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 19 – MTI** Multicast Hash Event Enable

| Value | Description |
|---|---|
| 0 | Wake on LAN multicast hash Event disabled |
| 1 | Wake on LAN multicast hash Event enabled |

**Bit 18 – SA1** Specific Address Register 1 Event Enable

| Value | Description |
|---|---|
| 0 | Wake on Specific Address Register 1 Event disabled |
| 1 | Wake on Specific Address Register 1 Event enabled |

**Bit 17 – ARP** ARP Request Event Enable

| Value | Description |
|---|---|
| 0 | Wake on LAN ARP request Event disabled |
| 1 | Wake on LAN ARP request Event enabled |

**Bit 16 – MAG** Magic Packet Event Enable

| Value | Description |
|---|---|
| 0 | Wake on LAN magic packet Event disabled |
| 1 | Wake on LAN magic packet Event enabled |

**Bits 15:0 – IP[15:0]** ARP Request IP Address
Wake on LAN ARP request IP address. Written to define the 16 least significant bits of the target IP address that is matched to generate a Wake on LAN event.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0x0000 | No Event generated, even if matched by the received frame. |
| 0x0001-0xFFFF | Wake on LAN Event generated for matching LSB of the target IP address. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 24.9.26. GMAC IPG Stretch Register

**Name:** IPGS
**Offset:** 0x0BC
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FL[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – FL[15:0]** Frame Length

Bits FL[7:0] are multiplied with the previously transmitted frame length (including preamble), and divided by FL[15:8]+1 (adding 1 to prevent division by zero). $\mathrm{RESULT} = \frac{\mathrm{FL[7:0]}}{\mathrm{F[15+8]+1}}$

If RESULT > 96 and the IP Stretch Enable bit in the Network Configuration Register (NCFGR.IPGSEN) is written to '1', RESULT is used for the transmit inter-packet-gap.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.27. GMAC Stacked VLAN Register

**Name:** SVLAN
**Offset:** 0x0C0
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ESVLAN | | | | | | | |
| Access | - | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | VLAN_TYPE[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VLAN_TYPE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – ESVLAN** Enable Stacked VLAN Processing Mode
0: Disable the stacked VLAN processing mode
1: Enable the stacked VLAN processing mode

| Value | Description |
|---|---|
| 0 | Stacked VLAN Processing disabled |
| 1 | Stacked VLAN Processing enabled |

**Bits 15:0 – VLAN_TYPE[15:0]** User Defined VLAN_TYPE Field
When Stacked VLAN is enabled (ESVLAN=1), the first VLAN tag in a received frame will only be accepted if the VLAN type field is equal to this user defined VLAN_TYPE, OR equal to the standard VLAN type (0x8100).
**Note:** The second VLAN tag of a Stacked VLAN packet will only be matched correctly if its VLAN_TYPE field equals 0x8100.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.28. GMAC Transmit PFC Pause Register

**Name:** TPFCP
**Offset:** 0x0C4
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PQ[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PEV[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:8 – PQ[7:0]** Pause Quantum
If NCR.TXPBPF =1, then for each entry equal to zero in the Transmit PFC Pause Register[15:8], the PFC pause frame's pause quantum field associated with that entry will be taken from the Transmit Pause Quantum Register. For each entry equal to one in the Transmit PFC Pause Register [15:8], the pause quantum associated with that entry will be zero.

**Bits 7:0 – PEV[7:0]** Priority Enable Vector
If NCR.TXPBPF =1, then the priority enable vector of the PFC priority based pause frame will be set equal to the value stored in this register [7:0].


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 24.9.29. GMAC Specific Address 1 Mask Bottom

**Name:** SAMB1
**Offset:** 0x0C8
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADDR[31:0]** Specific Address 1 Mask
Setting a bit to '1' masks the corresponding bit in the Specific Address 1 Bottom register (SAB1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.30. GMAC Specific Address Mask 1 Top

**Name:** SAMT1
**Offset:** 0x0CC
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – ADDR[15:0]** Specific Address 1 Mask
Setting a bit to '1' masks the corresponding bit in the Specific Address 1 register SAT1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.31. GMAC 1588 Timer Nanosecond Comparison Register

**Name:** NSC
**Offset:** 0x0DC
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | NANOSEC[20:16] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NANOSEC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NANOSEC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 20:0 – NANOSEC[20:0]** 1588 Timer Nanosecond Comparison Value
Value is compared to the bits [45:24] of the TSU timer count value (upper 21 bits of nanosecond value).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.32. GMAC 1588 Timer Second Comparison Low Register

**Name:** SCL
**Offset:** 0x0E0
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SEC[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SEC[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SEC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SEC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – SEC[31:0]** 1588 Timer Second Comparison Value
Value is compared to seconds value bits [31:0] of the TSU timer count value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.33. GMAC 1588 Timer Second Comparison High Register

**Name:** SCH
**Offset:** 0x0E4
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SEC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SEC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – SEC[15:0]** 1588 Timer Second Comparison Value
Value is compared to the top 16 bits (most significant 16 bits [47:32] of seconds value) of the TSU timer count value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.34. GMAC PTP Event Frame Transmitted Seconds High Register

**Name:** EFTSH
**Offset:** 0x0E8
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RUD[15:0]** Register Update
The register is updated with the value that the IEEE 1588 timer seconds register held when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.35. GMAC PTP Event Frame Received Seconds High Register

**Name:** EFRSH
**Offset:** 0x0EC
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RUD[15:0]** Register Update

The register is updated with the value that the IEEE 1588 timer seconds register held when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.36. GMAC PTP Peer Event Frame Transmitted Seconds High Register

**Name:** PEFTSH
**Offset:** 0x0F0
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RUD[15:0]** Register Update

The register is updated with the value that the IEEE 1588 timer seconds register held when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.37. GMAC PTP Peer Event Frame Received Seconds High Register

**Name:** PEFRSH
**Offset:** 0x0F4
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RUD[15:0]** Register Update
The register is updated with the value that the 1588 timer seconds register held when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.38. GMAC Octets Transmitted Low Register

**Name:** OTLO
**Offset:** 0x100
**Reset:** 0x00000000
**Property:** Read-Only

When reading the Octets Transmitted and Octets Received Registers, bits [31:0] should be read prior to bits [47:32] to ensure reliable operation.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TXO[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TXO[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TXO[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TXO[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TXO[31:0]** Transmitted Octets
Transmitted octets in valid frames of any type without errors, bits [31:0]. This counter is 48-bits, and is read through two registers. This count does not include octets from automatically generated pause frames.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 24.9.39. GMAC Octets Transmitted High Register

**Name:** OTHI
**Offset:** 0x104
**Reset:** 0x00000000
**Property:** Read-Only

When reading the Octets Transmitted and Octets Received Registers, bits [31:0] should be read prior to bits [47:32] to ensure reliable operation.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TXO[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TXO[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – TXO[15:0]** Transmitted Octets
Transmitted octets in valid frames of any type without errors, bits [47:32]. This counter is 48-bits, and is read through two registers. This count does not include octets from automatically generated pause frames.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.40. GMAC Frames Transmitted

**Name:** FT
**Offset:** 0x108
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | FTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – FTX[31:0]** Frames Transmitted without Error
Frames transmitted without error. This register counts the number of frames successfully transmitted, i.e., no underrun and not too many retries. Excludes pause frames.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.41. GMAC Broadcast Frames Transmitted Register

**Name:** BCFT
**Offset:** 0x10C
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – BFTX[31:0]** Broadcast Frames Transmitted without Error
This register counts the number of broadcast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.42. GMAC Multicast Frames Transmitted Register

**Name:** MFT
**Offset:** 0x110
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | MFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – MFTX[31:0]** Multicast Frames Transmitted without Error
This register counts the number of multicast frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.43. GMAC Pause Frames Transmitted Register

**Name:** PFT
**Offset:** 0x114
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – PFTX[15:0]** Pause Frames Transmitted Register
This register counts the number of pause frames transmitted. Only pause frames triggered by the register interface or through the external pause pins are counted as pause frames. Pause frames received through the FIFO interface are counted in the frames transmitted counter.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 24.9.44. GMAC 64 Byte Frames Transmitted Register

**Name:** BFT64
**Offset:** 0x118
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFTX[31:0]** 64 Byte Frames Transmitted without Error
This register counts the number of 64 byte frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.45. GMAC 65 to 127 Byte Frames Transmitted Register

**Name:** TBFT127
**Offset:** 0x11C
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFTX[31:0]** 65 to 127 Byte Frames Transmitted without Error
This register counts the number of 65 to 127 byte frames successfully transmitted without error, i.e., no underrun and not too many retries. Excludes pause frames.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.46. GMAC 128 to 255 Byte Frames Transmitted Register

**Name:** TBFT255
**Offset:** 0x120
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFTX[31:0]** 128 to 255 Byte Frames Transmitted without Error
This register counts the number of 128 to 255 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.47. GMAC 256 to 511 Byte Frames Transmitted Register

**Name:** TBFT511
**Offset:** 0x124
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFTX[31:0]** 256 to 511 Byte Frames Transmitted without Error
This register counts the number of 256 to 511 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.48. GMAC 512 to 1023 Byte Frames Transmitted Register

**Name:** TBFT1023
**Offset:** 0x128
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFTX[31:0]** 512 to 1023 Byte Frames Transmitted without Error
This register counts the number of 512 to 1023 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.49. GMAC 1024 to 1518 Byte Frames Transmitted Register

**Name:** TBFT1518
**Offset:** 0x12C
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFTX[31:0]** 1024 to 1518 Byte Frames Transmitted without Error
This register counts the number of 1024 to 1518 byte frames successfully transmitted without error, i.e., no underrun and not too many retries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.50. GMAC Greater Than 1518 Byte Frames Transmitted Register

**Name:** GTBFT1518
**Offset:** 0x130
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFTX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFTX[31:0]** Greater than 1518 Byte Frames Transmitted without Error
This register counts the number of 1518 or above byte frames successfully transmitted without error i.e., no underrun and not too many retries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.51. GMAC Transmit Underruns Register

**Name:** TUR
**Offset:** 0x134
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | TXUNR[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TXUNR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – TXUNR[9:0]** Transmit Underruns
This register counts the number of frames not transmitted due to a transmit underrun. If this register is incremented then no other statistics register is incremented.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.52. GMAC Single Collision Frames Register

**Name:** SCF
**Offset:** 0x138
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | SCOL[17:16] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SCOL[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SCOL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 17:0 – SCOL[17:0]** Single Collision
This register counts the number of frames experiencing a single collision before being successfully transmitted i.e., no underrun.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.53. GMAC Multiple Collision Frames Register

**Name:** MCF
**Offset:** 0x13C
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | MCOL[17:16] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MCOL[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

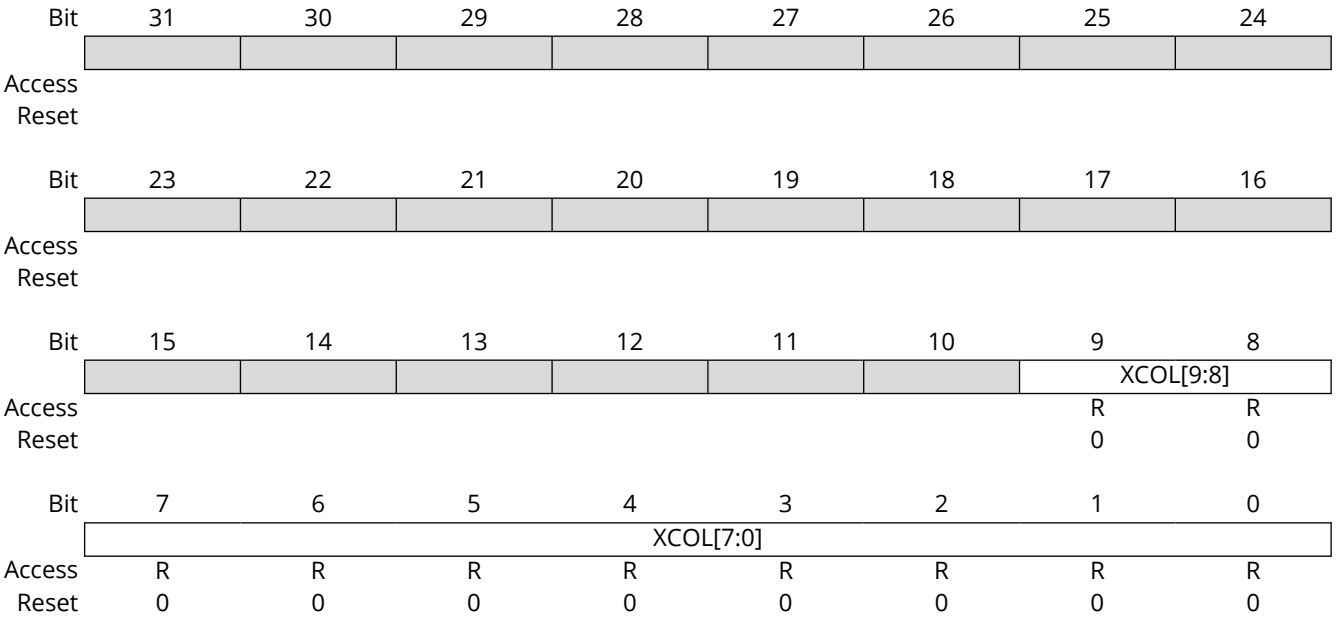
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MCOL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 17:0 – MCOL[17:0]** Multiple Collision
This register counts the number of frames experiencing between two and fifteen collisions prior to being successfully transmitted, i.e., no underrun and not too many retries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.54. GMAC Excessive Collisions Register

**Name:** EC
**Offset:** 0x140
**Reset:** 0x00000000
**Property:** Read-only



| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | XCOL[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | XCOL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – XCOL[9:0]** Excessive Collisions
This register counts the number of frames that failed to be transmitted because they experienced 16 collisions.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.55. GMAC Late Collisions Register

**Name:** LC
**Offset:** 0x144
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | LCOL[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LCOL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – LCOL[9:0]** Late Collisions
This register counts the number of late collisions occurring after the slot time (512 bits) has expired. In 10/100 mode, late collisions are counted twice i.e., both as a collision and a late collision.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.56. GMAC Deferred Transmission Frames Register

**Name:** DTF
**Offset:** 0x148
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DEFT[17:16] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DEFT[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DEFT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 17:0 – DEFT[17:0]** Deferred Transmission

This register counts the number of frames experiencing deferral due to carrier sense being active on their first attempt at transmission. Frames involved in any collision are not counted nor are frames that experienced a transmit underrun.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.57. GMAC Carrier Sense Errors Register

**Name:** CSE
**Offset:** 0x14C
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CSR[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CSR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – CSR[9:0]** Carrier Sense Error

This register counts the number of frames transmitted with carrier sense was not seen during transmission or where carrier sense was de-asserted after being asserted in a transmit frame without collision (no underrun). Only incremented in half duplex mode. The only effect of a carrier sense error is to increment this register. The behavior of the other statistics registers is unaffected by the detection of a carrier sense error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.58. GMAC Octets Received Low Register

**Name:** ORLO
**Offset:** 0x150
**Reset:** 0x00000000
**Property:** Read-Only

When reading the Octets Transmitted and Octets Received Registers, bits [31:0] should be read prior to bits [47:32] to ensure reliable operation.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RXO[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RXO[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RXO[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXO[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RXO[31:0]** Received Octets
Received octets in frame without errors [31:0]. The number of octets received in valid frames of any type. This counter is 48-bits and is read through two registers. This count does not include octets from pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.59. GMAC Octets Received High Register

**Name:** ORHI
**Offset:** 0x154
**Reset:** 0x00000000
**Property:** Read-only

When reading the Octets Transmitted and Octets Received Registers, bits 31:0 should be read prior to bits 47:32 to ensure reliable operation.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RXO[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXO[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RXO[15:0]** Received Octets
Received octets in frame without errors [47:32]. The number of octets received in valid frames of any type. This counter is 48-bits and is read through two registers. This count does not include octets from pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.60. GMAC Frames Received Register

**Name:** FR
**Offset:** 0x158
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | FRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – FRX[31:0]** Frames Received without Error
This bit field counts the number of frames successfully received, excluding pause frames. It is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.61. GMAC Broadcast Frames Received Register

**Name:** BCFR
**Offset:** 0x15C
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – BFRX[31:0]** Broadcast Frames Received without Error
Broadcast frames received without error. This bit field counts the number of broadcast frames successfully received. This excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.62. GMAC Multicast Frames Received Register

**Name:** MFR
**Offset:** 0x160
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | MFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – MFRX[31:0]** Multicast Frames Received without Error
This register counts the number of multicast frames successfully received without error, excluding pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.63. GMAC Pause Frames Received Register

**Name:** PFR
**Offset:** 0x164
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – PFRX[15:0]** Pause Frames Received Register
This register counts the number of pause frames received without error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.64. GMAC 64 Byte Frames Received Register

**Name:** BFR64
**Offset:** 0x168
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFRX[31:0]** 64 Byte Frames Received without Error
This bit field counts the number of 64 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.65. GMAC 65 to 127 Byte Frames Received Register

**Name:** TBFR127
**Offset:** 0x16C
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFRX[31:0]** 65 to 127 Byte Frames Received without Error
This bit field counts the number of 65 to 127 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.66. GMAC 128 to 255 Byte Frames Received Register

**Name:** TBFR255
**Offset:** 0x170
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFRX[31:0]** 128 to 255 Byte Frames Received without Error
This bit field counts the number of 128 to 255 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.67. GMAC 256 to 511 Byte Frames Received Register

**Name:** TBFR511
**Offset:** 0x174
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFRX[31:0]** 256 to 511 Byte Frames Received without Error
This bit fields counts the number of 256 to 511 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.68. GMAC 512 to 1023 Byte Frames Received Register

**Name:** TBFR1023
**Offset:** 0x178
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFRX[31:0]** 512 to 1023 Byte Frames Received without Error
This bit field counts the number of 512 to 1023 byte frames successfully received without error. Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.69. GMAC 1024 to 1518 Byte Frames Received Register

**Name:** TBFR1518
**Offset:** 0x17C
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFRX[31:0]** 1024 to 1518 Byte Frames Received without Error
This bit field counts the number of 1024 to 1518 byte frames successfully received without error, i.e., no underrun and not too many retries.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.70. GMAC 1519 to Maximum Byte Frames Received Register

**Name:** TMXBFR
**Offset:** 0x180
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – NFRX[31:0]** 1519 to Maximum Byte Frames Received without Error
This bit field counts the number of 1519 Byte or above frames successfully received without error. Maximum frame size is determined by the Maximum Frame Size bit (MAXFS, 1536 Bytes) or Jumbo Frame Size bit (JFRAME, 10240 Bytes) in the Network Configuration Register (NCFGR). Excludes pause frames, and is only incremented if the frame is successfully filtered and copied to memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.71. GMAC Undersized Frames Received Register

**Name:** UFR
**Offset:** 0x184
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | UFRX[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | UFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – UFRX[9:0]** Undersize Frames Received
This bit field counts the number of frames received less than 64 bytes in length (10/100 mode, full duplex) that do not have either a CRC error or an alignment error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.72. GMAC Oversized Frames Received Register

**Name:** OFR
**Offset:** 0x188
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | OFRX[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OFRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – OFRX[9:0]** Oversized Frames Received
This pit field counts the number of frames received exceeding 1518 Bytes in length (1536 Bytes if NCFGR.MAXFS is written to '1') but do not have either a CRC error, an alignment error, nor a receive symbol error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.73. GMAC Jabbers Received Register

**Name:** JR
**Offset:** 0x18C
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | JRX[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | JRX[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – JRX[9:0]** Jabbers Received
This bit field counts the number of frames received exceeding 1518 Bytes in length (1536 Bytes if NCFGR.MAXFS is written to '1') and have either a CRC error, an alignment error or a receive symbol error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.74. GMAC Frame Check Sequence Errors Register

**Name:** FCSE
**Offset:** 0x190
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | FCKR[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FCKR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – FCKR[9:0]** Frame Check Sequence Errors

The register counts frames that are an integral number of bytes, have bad CRC and are between 64 and 1518 bytes in length (1536 Bytes if NCFGR.MAXFS is written to '1'). This register is also incremented if a symbol error is detected and the frame is of valid length and has an integral number of bytes.

This register is incremented for a frame with bad FCS, regardless of whether it is copied to memory due to ignore FCS mode (enabled by writing NCFGR.IRXFCS=1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.75. GMAC Length Field Frame Errors Register

**Name:** LFFE
**Offset:** 0x194
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | LFER[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LFER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – LFER[9:0]** Length Field Frame Errors
This bit field counts the number of frames received that have a measured length shorter than that extracted from the length field (Bytes 13 and 14). This condition is only counted if the value of the length field is less than 0x0600, the frame is not of excessive length and checking is enabled by writing a '1' to the Length Field Error Frame Discard bit in the Network Configuration Register (GMAC_NCFGR.LFERD).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.76. GMAC Receive Symbol Errors Register

**Name:** RSE
**Offset:** 0x198
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RXSE[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXSE[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – RXSE[9:0]** Receive Symbol Errors
This bit field counts the number of frames that had GRXER asserted during reception. For 10/100 mode symbol errors are counted regardless of frame length checks. Receive symbol errors will also be counted as an FCS or alignment error if the frame is between 64 and 1518 Bytes (1536 Bytes if NCFGR.MAXFS=1). If the frame is larger it will be recorded as a jabber error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.77. GMAC Alignment Errors Register

**Name:** AE
**Offset:** 0x19C
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | AER[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | AER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – AER[9:0]** Alignment Errors
This bit field counts the frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of Bytes and are between 64 and 1518 Bytes in length (1536 if NCFGR.MAXFS=1). This register is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of bytes.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.78. GMAC Receive Resource Errors Register

**Name:** RRE
**Offset:** 0x1A0
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RXRER[17:16] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RXRER[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXRER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 17:0 – RXRER[17:0]** Receive Resource Errors

This bit field counts frames that are not an integral number of bytes long and have bad CRC when their length is truncated to an integral number of Bytes and are between 64 and 1518 Bytes in length (1536 if NCFGR.MAXFS=1). This bit field is also incremented if a symbol error is detected and the frame is of valid length and does not have an integral number of Bytes.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.79. GMAC Receive Overruns Register

**Name:** ROE
**Offset:** 0x1A4
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RXOVR[9:8] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXOVR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:0 – RXOVR[9:0]** Receive Overruns
This bit field counts the number of frames that are address recognized but were not copied to memory due to a receive overrun.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.80. GMAC IP Header Checksum Errors Register

**Name:** IHCE
**Offset:** 0x1A8
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HCKER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – HCKER[7:0]** IP Header Checksum Errors
This register counts the number of frames discarded due to an incorrect IP header checksum, but are between 64 and 1518 Bytes (1536 Bytes if GMAC_NCFGR.MAXFS=1) and do not have a CRC error, an alignment error, nor a symbol error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.81. GMAC TCP Checksum Errors Register

**Name:** TCE
**Offset:** 0x1AC
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCKER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – TCKER[7:0]** TCP Checksum Errors
This register counts the number of frames discarded due to an incorrect TCP checksum, but are between 64 and 1518 Bytes (1536 Bytes if NCFGR.MAXFS=1) and do not have a CRC error, an alignment error, nor a symbol error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.82. GMAC UDP Checksum Errors Register

**Name:** UCE
**Offset:** 0x1B0
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | UCKER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – UCKER[7:0]** UDP Checksum Errors
This register counts the number of frames discarded due to an incorrect UDP checksum, but are between 64 and 1518 Bytes (1536 Bytes if NCFGR.MAXFS=1) and do not have a CRC error, an alignment error, nor a symbol error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.83. GMAC 1588 Timer Increment Sub-nanoseconds Register

**Name:** TISUBN
**Offset:** 0x1BC
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | LSBTIR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LSBTIR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – LSBTIR[15:0]** Lower Significant Bits of Timer Increment Register
Lower significant bits of Timer Increment Register [15:0], giving a 24-bit timer_increment counter. These bits are the sub-ns value which the 1588 timer will be incremented each clock cycle. Bit n = $2^{(n-16)}$ ns giving a resolution of approximately $15.2E^{-15}$ sec.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.84. GMAC 1588 Timer Seconds High Register

**Name:** TSH
**Offset:** 0x1C0
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TCS[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – TCS[15:0]** Timer Count in Seconds
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.85. GMAC 1588 Timer Seconds Low Register

**Name:** TSL
**Offset:** 0x1D0
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TCS[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TCS[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TCS[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TCS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TCS[31:0]** Timer Count in Seconds
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.86. 1588 Timer Sync Strobe Seconds [31:0] Register

**Name:** TSSSL
**Offset:** 0x1C8
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | VTS[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | VTS[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | VTS[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VTS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – VTS[31:0]** Value of Timer Seconds Register Capture
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.87. GMAC 1588 Timer Sync Strobe Nanoseconds Register

**Name:** TSSN
**Offset:** 0x1CC
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | VTN[29:24] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | VTN[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | VTN[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VTN[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:0 – VTN[29:0]** Value Timer Nanoseconds Register Capture
This register is writable. It increments by 1 when the IEEE 1588 nanoseconds counter counts to one second. It may also be incremented when the Timer Adjust Register is written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.88. GMAC 1588 Timer Nanoseconds Register

**Name:** TN
**Offset:** 0x1D4
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | TNS[29:24] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TNS[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TNS[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TNS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:0 – TNS[29:0]** Timer Count in Nanoseconds
This register is writable. It can also be adjusted by writes to the IEEE 1588 Timer Adjust Register. It increments by the value of the IEEE 1588 Timer Increment Register each clock cycle.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.89. GMAC 1588 Timer Adjust Register

**Name:** TA
**Offset:** 0x1D8
**Reset:** 0x00000000
**Property:** Write-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADJ | | ITDT[29:24] | | | | | |
| Access | W | | W | W | W | W | W | W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ITDT[23:16] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ITDT[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ITDT[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – ADJ** Adjust 1588 Timer
Write as '1' to subtract from the 1588 timer. Write as '0' to add to it.

**Bits 29:0 – ITDT[29:0]** Increment/Decrement
The number of nanoseconds to increment or decrement the IEEE 1588 Timer Nanoseconds Register. If necessary, the IEEE 1588 Seconds Register will be incremented or decremented.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.90. GMAC IEEE 1588 Timer Increment Register

**Name:** TI
**Offset:** 0x1DC
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NIT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ACNS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CNS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – NIT[7:0]** Number of Increments
The number of increments after which the alternative increment is used.

**Bits 15:8 – ACNS[7:0]** Alternative Count Nanoseconds
Alternative count of nanoseconds by which the 1588 Timer Nanoseconds Register will be incremented each clock cycle.

**Bits 7:0 – CNS[7:0]** Count Nanoseconds
A count of nanoseconds by which the IEEE 1588 Timer Nanoseconds Register will be incremented each clock cycle.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.91. GMAC PTP Event Frame Transmitted Seconds Low Register

**Name:** EFTSL
**Offset:** 0x1E0
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RUD[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RUD[31:0]** Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.92. GMAC PTP Event Frame Transmitted Nanoseconds Register

**Name:** EFTN
**Offset:** 0x1E4
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | RUD[29:24] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:0 – RUD[29:0]** Register Update
The register is updated with the value that the IEEE 1588 Timer Nanoseconds Register holds when the SFD of a PTP transmit primary event crosses the MII interface. An interrupt is issued when the bit field is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.93. GMAC PTP Event Frame Received Seconds Low Register

**Name:** EFRSL
**Offset:** 0x1E8
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RUD[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RUD[31:0]** Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.94. GMAC PTP Event Frame Received Nanoseconds Register

**Name:** EFRN
**Offset:** 0x1EC
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | RUD[29:24] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:0 – RUD[29:0]** Register Update
The register is updated with the value that the IEEE 1588 Timer Nanoseconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.95. GMAC PTP Peer Event Frame Transmitted Seconds Low Register

**Name:** PEFTSL
**Offset:** 0x1F0
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RUD[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RUD[31:0]** Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.96. GMAC PTP Peer Event Frame Transmitted Nanoseconds Register

**Name:** PEFTN
**Offset:** 0x1F4
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | RUD[29:24] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:0 – RUD[29:0]** Register Update
The register is updated with the value that the 1588 Timer Nanoseconds Register holds when the SFD of a PTP transmit peer event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.97. GMAC PTP Peer Event Frame Received Seconds Low Register

**Name:** PEFRSL
**Offset:** 0x1F8
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RUD[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RUD[31:0]** Register Update
The register is updated with the value that the IEEE 1588 Timer Seconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.98. GMAC PTP Peer Event Frame Received Nanoseconds Register

**Name:** PEFRN
**Offset:** 0x1FC
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | RUD[29:24] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUD[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUD[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUD[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:0 – RUD[29:0]** Register Update

The register is updated with the value that the IEEE 1588 Timer Nanoseconds Register holds when the SFD of a PTP receive primary event crosses the MII interface. An interrupt is issued when the register is updated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.99. Received LPI Transitions

**Name:** RLPITR
**Offset:** 0x270
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RLPITR[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RLPITR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RLPITR[15:0]** Received LPI Transitions
The value of this bit field is a counter of transitions from receiving normal idle to receiving low power idle.
Cleared on read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.100. Received LPI Time

**Name:** RLPITI
**Offset:** 0x274
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RLPITI[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RLPITI[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RLPITI[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:0 – RLPITI[23:0]** Received LPI Time
The value of this bit field increments once every 16 AHB clock cycles when the Low Power Idle Enable bit in the Network Configuration Register (NCR.LPI) is written to '1'.
Cleared on read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.101. Transmit LPI Transitions

**Name:** TLPITR
**Offset:** 0x278
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TLPITR[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TLPITR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – TLPITR[15:0]** Transmit LPI Transitions
A count of the number of times the Low Power Idle Enable bit in the Network Configuration Register (NCR.LPI) goes from '0' to '1'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 24.9.102. Transmit LPI Time

**Name:** TLPITI
**Offset:** 0x27C
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TLPITI[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TLPITI[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TLPITI[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:0 – TLPITI[23:0]** Transmit LPI Time
The value of this bit field increments once every 16 AHB clock cycles when the Low Power Idle Enable bit in the Network Configuration Register (NCR.LPI) is written to '1'.
Cleared on read.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 25. NVMCTRL – Nonvolatile Memory Controller

## 25.1. Overview

Non-volatile memory (NVM) is a reprogrammable Flash memory that retains program and data storage, even when powered off. The NVM Controller (NVMCTRL) embeds two banks: one bank can be read while the other is programmed (RWW). It is connected to the AHB and APB bus interfaces for system access to the NVM block. The AHB interfaces are used for reads and writes to the NVM block, while the APB interface is used for commands and configuration.

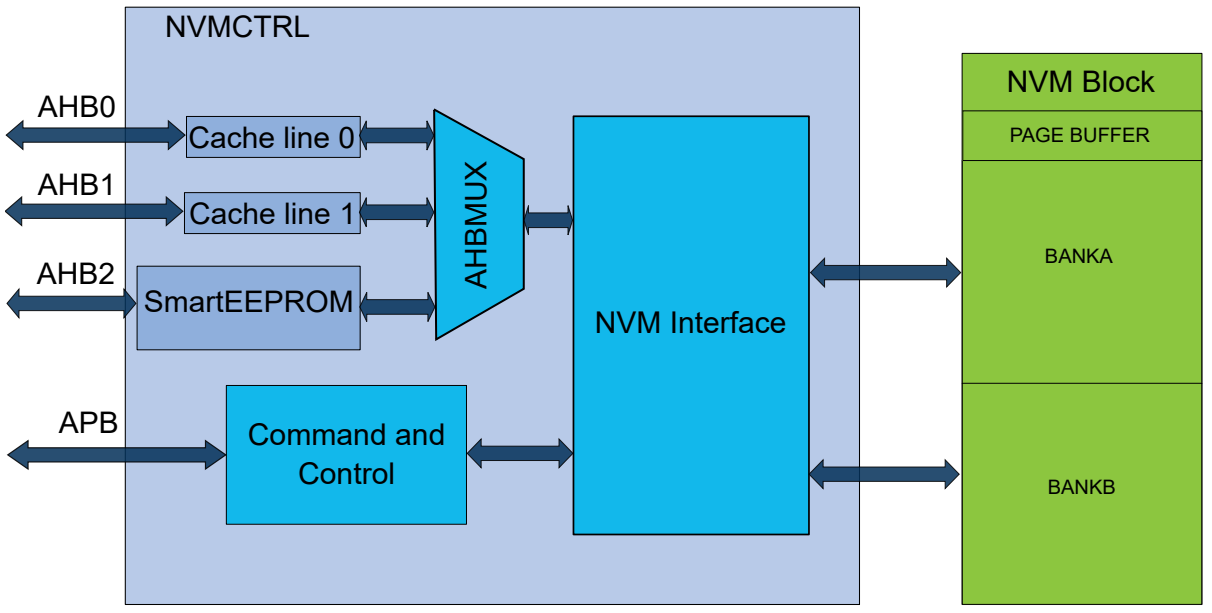
## 25.2. Features

The following are key features of the NVMCTRL:

- Two 32-bit AHB interfaces for reads and writes in the NVM main address space
- SmartEEPROM (integrated EEPROM emulation algorithm)
- Read-while-write (Any bank can be read while programming the other one)
- All NVM sections are memory mapped to the AHB, including calibration and system configuration
- 32-bit APB interface for commands and control
- Programmable wait states for read optimization
- Regions can be individually protected or unprotected
- Additional protection for bootloader
- Supports device protection through a security bit
- Interface to Power Manager to power-down Flash blocks while in sleep modes
- Can optionally wake up on exit from sleep or on first access
- Single line cache per AHB interface
- Dual bank for safer application upgrade
- Error Correction Code (ECC)

## 25.3. Block Diagram

**Figure 25-1.** Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 25.4. Signal Description

Not applicable.

## 25.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described in the following sections.

### 25.5.1. Power Management

The NVMCTRL will continue to operate in any sleep mode where the selected source clock is running. The NVMCTRL interrupts can be used to wake up the device from sleep modes.

The NVM block can be put into a low-power mode either automatically when the Power Manager enters standby mode, or when the SPRM command is issued. The NVMCTRL can wake-up when the Power Manager leaves sleep mode or on AHB access or when a command requires the NVM to be active. This is based on the Control A register (CTRLA) PRM bit setting. Read the CTRLA register description for more details.

NVM wake-up time can be traded with static power consumption depending on the PM STDBYCFG.FASTWKUP setting.

**Related Links**

PM – Power Manager

### 25.5.2. Clocks

Two synchronous clocks are used by the NVMCTRL. One is provided by the AHB bus (CLK_NVMCTRL_AHB) and the other is provided by the APB bus (CLK_NVMCTRL_APB). When changing the AHB bus frequency, the user must ensure that the NVM Controller is configured with the proper number of wait states. Refer to the Electrical Characteristics for the exact number of wait states to be used for a particular frequency range. Automatic wait state generation can be use by setting the Auto Wait State bit in the Control A register (NVMCTRL.CTRLA.AUTOWS). Alternatively a custom programmable number of wait states can be set by writing the NVM Read Wait State bits (NVMCTRL.CTRLA.RWS) to optimize performance.

**Related Links**

CTRLA

Control A

### 25.5.3. DMA

The NVMCTRL supports AHB burst transfers. It is possible to write the page buffer in sequence without AHB rearbitration in case of concurrent AHB writes to the page buffer to guarantee data integrity.

### 25.5.4. Interrupts

The NVM Controller interrupt request line is connected to the interrupt controller. Using the NVMCTRL interrupt requires the interrupt controller to be programmed first.

### 25.5.5. Debug Operation

When the CPU is halted in debug mode, the ECC feature of the NVMCTRL will correct and log ECC errors based on the table below.

**Table 25-1.** ECC Debug Operation

| DBGCTRL.ECCELOG | DBGCTRL.ECCDIS | DBGCTRL.ECCDIS |
|---|---|---|
| 0 | 0 | ECC errors from debugger reads are corrected, but not logged in INTFLAG. |
| 1 | 0 | ECC errors from debugger reads are corrected and logged in INTFLAG. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 25-1.** ECC Debug Operation (continued)

| DBGCTRL.ECCELOG | DBGCTRL.ECCDIS | DBGCTRL.ECCDIS |
|---|---|---|
| X | 1 | ECC errors from debugger reads are neither corrected nor logged in INTFLAG. |

Reading the SmartEEPROM configured in buffered mode with a debugger is intrusive, since the pagebuffer must be flushed when the read is performed in a page under modification.

Access to the NVM block can be protected by the security bit. In this case, the NVM block will not be accessible. See the section on the NVMCTRL Security Bit for details.

### 25.5.6. Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the Interrupt Flag Status and Clear register (INTFLAG).

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

**Related Links**
PAC - Peripheral Access Controller

### 25.5.7. Analog Connections

Not applicable.

## 25.6. Functional Description

### 25.6.1. Principle of Operation

The NVM Controller is a Client on the AHB (AHB0, AHB1 and AHB2) and APB buses. It responds to commands, read requests, and write requests based on user configuration. AHB0 and AHB1 allow access to the NVM main address space, the auxiliary space, and the page buffer. AHB2 provides access to the SmartEEPROM interface that indirectly accesses the reserved area in the NVM for EEPROM emulation.

#### 25.6.1.1. Initialization

After power-up, the NVM Controller goes through a power-up sequence. During this time, access to the NVM Controller from the AHB bus is halted. Upon power-up completion, the NVM Controller is operational without any need for user configuration.

#### 25.6.1.2. Software Reset

Software reset is triggered by the SWRST command, and does the following:

- NVM (physical memory) reset
- Device power-up sequence (redo the device calibration)
- Reset all APB configuration registers (and status)

**Note:**
STATUS.READY goes low when the SWRST command starts to execute.

STATUS.READY goes high when the SWRST command has completed.

Any AHB0/1/2 access is stalled until the command has completed.

### 25.6.2. Memory Organization

Memory space is divided as follows:

- The main address space where 2 physical NVM banks (BANKA and BANKB) are mapped.
- The auxiliary space which contains:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The user page (USER)
- The calibration page (CB)
- Factory and signature pages (FS)

The BANKA and BANKB can be swapped in the address space. For additional information, refer to the Memory Bank Swapping.

Refer to the Physical Memory Map for memory sizes and addresses for each device.

The BANKA, BANKB and AUX pages have different erase and write granularities, as provided in the table below.

**Table 25-2.** Erase and Write Granularity

| | Erase Granularity | Write Granularity |
|---|---|---|
| BANKA | Block | Quad-Word or Page |
| BANKB | Block | Quad-Word or Page |
| AUX | Page | Quad-Word |

The NVM is organized into two banks, each bank is organized into blocks, where each block contains sixteen pages and page size is 512 bytes.

The lower blocks in the NVM main address space can be allocated as a boot loader section by using the BOOTPROT fuses, and the upper rows can be allocated to SmartEEPROM.

The NVM memory is separated into six parts:

- CB space: Contains factory calibration and system configuration information.
  - Address; 0x00800000
  - Size: 1 page
  - Property: Read-Only
- FS space: Contains the factory signature information.
  - Address; 0x00806000
  - Size: 4 pages
  - Property: Read-Only.
- USER space: Contains user-defined startup configuration. The first word is reserved, and used during the NVMCTRL start-up to automatically configure the device.
  - Address: 0x00804000
  - Size: 1 page
  - Property: Read-Write
- Main address space: Divided into 32 equal size regions. Each region can be protected against write or erase operation. The 32-bit RUNLOCK register reflects the protection of each region. This register is automatically updated after power-up with the region lock user fuse data found in the NVM LOCKS field, refer to the Chapter Memories. To lock or unlock a region, the Lock Region (LR) or Unlock Regions (UR) commands can be issued.
  - Address: 0x00000000
  - Size: PARAM.NVMP pages.
  - Property: Read-Write
- Bootloader space
  The bootloader section starts at the beginning of the main address space; Its size is defined by the NVM User Row BOOTPROT[3:0] fuse. It is protected against write or erase operations, except if STATUS.BPDIS is set. Issuing a write or erase command at an address inside the BOOTPROT


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

section sets STATUS.PROGE and STATUS.LOCKE. STATUS.BPDIS can be set by issuing the Set BOOTPROT Disable command (SBPDIS). It is cleared by issuing the Clear BOOTPROT Disable command (CBPDIS). This allows to program a new bootloader without changing the user page and issuing a new NVMCTRL startup sequence to reload the user configuration. The BOOTPROT section is not erased during a Chip-Erase operation even if STATUS.BPDIS is high.
  - Address: 0x00000000
  - Size: (15 - STATUS.BOOTPROT) × 8192
  - Property: Read-Only.
- SmartEEPROM raw data space: The SmartEEPROM algorithm emulates an EEPROM with a portion of the NVM main. SmartEEPROM raw data is mapped at the end of the main address space. SmartEEPROM allocated space in the main address space is not accessible from AHB0/1. Any AHB access throws a Hardfault exception. Any command issued with ADDR pointing in the SmartEEPPROM space is discarded, INTFLAG.DONE and INTFLAG.ADDRE are set in this case.
  - Address: PARAM.NVMP*512-2*SEESTAT.SBLK*8192
  - Size: 2*SEESTAT.SBLK*8192
  - Property: Not readable, not writable

Each section has a different protection status, as provided in the table below.

**Table 25-3.** Protection Status

| Section/Operation | Write protection | Erase protection | Chip-Erase protection |
|---|---|---|---|
| Bootloader | Yes | Yes | Yes |
| SmartEEPROM | Configurable | Configurable | No |
| Main Array | Configurable | Configurable | No |

**Related Links**
Physical Memory Map
DSU - Device Service Unit

### 25.6.3. Memory Bank Swapping

The two physical banks BANKA and BANKB are mapped in the NVM main address space and can be swapped. If STATUS.AFIRST contains '1', then BANKA is mapped to the NVM main address space Base Address, otherwise it is BANKB.

The start address of BANKA & BANKB depends on STATUS.AFIRST and on the size of the Flash. Refer to the Physical Memory Map for memory sizes and addresses for each device.

**Related Links**
Physical Memory Map

### 25.6.4. AHBMUX Arbitration

The AHBMUX arbitrates concurrent AHB0, AHB1 and SmartEEPROM accesses using a fixed priority scheme:

- AHB0 has the highest priority
- AHB1 has priority over SmartEEPROM
- SmartEEPROM has priority over AHB2

However, once a transfer has been accepted the AHB data phase must complete, meaning that a transaction can be stalled by a previously granted access with a lower priority. This can occur in Automatic Wait State mode or in Fixed Wait State mode when the Wait state is greater than zero.

AHBMUX does not re-arbitrate AHB burst transactions. This is useful in case of concurrent write transfers to the page buffer. If used in conjunction with the automatic write features (ADW, AQW,


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

APW) and if the burst transfer size is a multiple of the automatic write size, several Hosts can write the NVM without implementing any software semaphore checks.

It is possible to force the re-arbitration in case of burst transfers, as follows:

- on AHB0: by writing a '1' to CTRLA.AHBNS0
- on AHB1: by writing a '1' to CTRLA.AHBNS1

**Related Links**
High-Speed Bus System

### 25.6.5. Region Lock Bits

The NVM main address space is accessible through the AHB0 or AHB1 interfaces, and grouped into 32 equally sized regions regardless of BOOTPROT or SmartEEPROM settings. The region size is dependent on the flash memory size, and is given in the table below. Each region has a dedicated lock bit preventing writing and erasing pages in the region. After production, all regions will be unlocked.

**Table 25-4.** Region Size

| Memory Size [KB] | Region Size [KB] |
|---|---|
| 1024 | 32 |
| 512 | 16 |
| 256 | 8 |

To lock or unlock a region, the Lock Region and Unlock Region commands are provided. Writing one of these commands will temporarily lock/unlock the region containing the address loaded in the ADDR register. ADDR can be written by software, or the automatically loaded value from a AHB write operation can be used. The new setting will stay in effect until the next reset, or the setting can be changed again using the lock and unlock commands. The current status of the lock can be determined by reading the RUNLOCK register.

To change the default lock/unlock setting for a region, the user page must be written (See Memories chapter). Writing to the auxiliary space will take effect after the next reset. Therefore, a boot of the device is needed for changes in the lock/unlock setting to take effect. Refer to the Physical Memory Map for calibration and auxiliary space address mapping.

**Related Links**
Physical Memory Map

### 25.6.6. Command and Data Interface

The NVM Controller is addressable from the APB bus, while the NVM main address space is addressable from the AHB bus. Read and automatic page write operations are performed by addressing the NVM main address space directly, while other operations such as manual page writes and block erase must be performed by issuing commands through the NVM Controller.

To issue a command, the CTRLB.CMD bits must be written along with the CTRLB.CMDEX value (0xA5). STATUS.READY is cleared when a command is issued and set when it has completed. Any command written while STATUS.READY is low will be ignored causing INTFLAG.PROGE to rise. Refer to the CTRLB register description for more details.

Invalid commands are discarded and will set INTFLAG.PROGE and INTFLAG.DONE when issued.

The CTRLA register must be used to control the power reduction mode, read wait states, and the write mode.

Commands that require an address use the ADDR register as an argument. ADDR APB write access is locked by the NVMCTRL while being used internally. For instance if a write operation is started by the NVMCTRL, an APB write is discarded so that the write operation is performed at


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

the correct address. The discarded APB write is signaled by rising INTFLAG.ADDRE. Commands that needs an address will fail if issued while INTFLAG.ADDRE is set, such failure is signaled by rising INTFLAG.PROGE.

The APB ADDR register is updated upon:

- APB writes to the ADDR register address
- AHB writes to the page buffer

ADDR APB writes are discarded and report an INTFLAG.ADDRE error in the following cases:

- When written from APB while a command is reading it
- ADDR APB write access while writing the page buffer (AHB write): ADDR is written upon AHB writes and must stay valid until the page buffer has been written and also until automatic write command has been issued to the command interface when in automatic write mode (WMODE configured as ADW or AQW or AP)
- ADDR APB write access while the command interface reads it
- A command is executed at an illegal address

All commands that require an address are discarded when INTFLAG.ADDRE is set. INTFLAG.PROGE is set in this case. INTFLAG.ADDRE must be cleared before issuing such commands.

#### 25.6.6.1. NVM Read

Reading from the NVM main address space is performed via the AHB bus by addressing the NVM main address space or auxiliary address space directly. Read data is available after the number of read wait states has passed as configured in NVMCTRL.CTRLA.RWS.

The number of cycles data are delayed to the AHB bus is determined by the read wait states.

It is not possible to read two banks at the same time. In case of simultaneous read operations, transactions are arbitrated by the internal matrix. Arbitration scheme is fixed priority, AHB0 has the highest priority, AHB1 has priority over AHB2. In case of conflict, AHB interfaces with lower priority are stalled.

Reading in a bank stalls the bus when it is being programmed or erased except when the suspend feature is used.

Reading in a bank does not stall the bus when the other bank is being programmed or erased.

**Related Links**

Suspend/Resume

#### 25.6.6.2. NVM Write

The entire NVM main address space except the BOOTPROT section can be erased by a debugger Chip Erase command. Alternatively, blocks or pages can be individually erased using the Erase Page (EP) or Erase Block (EB) depending on the targeted address space. The NVM can be programmed using the Write Page (WP) or Write Quad Word (WQW) commands depending on the targeted address space. AHB writes automatically update the ADDR register. ADDR is write locked by the NVMCTRL until the pagebuffer write completes or until the appropriate write command has been passed to the command interface when in automatic write mode. Write commands are not supported in all address spaces, see the table below. These commands are detailed further in this section.

**Table 25-5.** Supported Commands Per Address Space

| | WP | WQW | EP | EB |
|---|---|---|---|---|
| Main Address Space | X | X | | X |
| User Page Address Space | | X | X | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Issuing an unsupported command on an address space sets the PROGE interrupt flag.

After programming the NVM main array, the region that the page resides in can be locked to prevent spurious write or erase sequences. Locking is performed on a per-region basis, and so locking a region locks all pages inside the region.

Data to be written to the NVM block is written through AHB and stored in an internal buffer called the page buffer. If the NVMCTRL is busy processing a write command (STATUS.READY=0) then the AHB bus is stalled upon an AHB write until the ongoing command completes. Writing the page buffer is allowed during a block erase operation. The page buffer contains the same number of bytes as an NVM page. Writes to the page buffer must be 32 bits. 16-bit or 8-bit writes to the page buffer is not allowed, and will cause a PAC error. Internally, writes to the page buffer are on a 64-bit basis through the page buffer load data registers (PBLDATA[1] and PBLDATA[0]). The PBLDATA register is a holding register for writes to the same 64-bit page buffer section. Data within a 64-bit section can be written in any order. Crossing a 64- bit boundary will reset the PBLDATA register to all ones. The following example assumes startup from reset where the current address is 0 and PBLDATA is all ones. Only 64 bits of the page buffer are written at a time, but 128 bits are shown for reference.

**Sequential 32-bit write example:**

- 32-bit 0x1 written to address 0
  - Page buffer[127:0] = {0xFFFFFFFF_FFFFFFFF, PBLDATA[63:32], 0x00000001}
  - PBLDATA[63:0] = {PBLDATA[63:32], 0x00000001}
- 32-bit 0x2 written to address 1
  - Page buffer[127:0] = {0xFFFFFFFF_FFFFFFFF, 0x00000002, PBLDATA[31:0]}
  - PBLDATA[63:0] = {0x00000002, PBLDATA[31:0]}
- 32-bit 0x3 written to address 2 (crosses 64-bit boundary)
  - Page buffer[127:0] = 0xFFFFFFFF_00000003_00000002_00000001
  - PBLDATA[63:0] = 0xFFFFFFFF_00000003

Random access writes to 32-bit words within the page buffer will overwrite the opposite word within the same 64-bit section with ones. In the following example, 0x00000001 is overwritten with 0xFFFFFFFF from the third write due to the 64-bit boundary crossing. Only 64 bits of the page buffer are written at a time, but 128 bits are shown for reference.

**Random access 32-bit AHB write example:**

- 32-bit 0x1 written to address 2
  - Page buffer[127:0] = 0xFFFFFFFF_00000001_FFFFFFFF_FFFFFFFF
  - PBLDATA[63:0] = 0xFFFFFFFF_00000001
- 32-bit 0x2 written to address 1
  - Page buffer[127:0] = 0xFFFFFFFF_00000001_00000002_FFFFFFFF
  - PBLDATA[63:0] = 0x00000002_FFFFFFFF
- 32-bit 0x3 written to address 3
  - Page buffer[127:0] = 0x00000003_FFFFFFFF_00000002_FFFFFFFF
  - PBLDATA[63:0] = 0x00000003_0xFFFFFFFF

BANKA and BANKB share the same page buffer. Writing to the NVM block through the AHB bus is buffered in the page buffer. For each AHB bus write, the address is stored in the ADDR register. After the page buffer has been loaded with the required number of bytes, the page can be written to the addressed location by setting CMD to Write Page to write the NVM main array and setting the key value to CMDEX. The LOAD bit in the STATUS register indicates whether the page buffer has been loaded or not. Before writing the page to memory, the accessed block must be erased.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Several write modes are supported and configured through CTRLA.WMODE:

- Manual (MAN): This is the default configuration. Because the address is automatically stored in ADDR during AHB write operations, the last given address will be present in the ADDR register. There is no need to load the ADDR register manually, unless a different page in memory is to be written. A write should be issued before writing to a different page.
- Automatic Write With Double Word Granularity (ADW): Automatically writes data with double-word granularity. In this case the WQW command is triggered at the quad-word addressed by ADDR when the last word in a double-word aligned block is written. The other double-word inside the page buffer must be all one. STATUS.READY goes low during the NVM write operation. INTFLAG.DONE flag is set upon completion.
- Automatic Write With Quad Word Granularity (AQW): Automatically writes data with quad-word granularity. In this case the WQW command is triggered at the quad-word addressed by ADDR when the last word in a quad-word aligned block is written. STATUS.READY goes low during the NVM write operation. The INTFLAG.DONE flag is set upon completion.
- Automatic Write With Page Granularity (AP): Automatically writes data with page granularity. In this case the WP command is triggered at the page addressed by ADDR when the last word in a page aligned block is written. STATUS.READY goes low during the NVM write operation. INTFLAG.DONE flag is set upon completion.

These write modes are supported for writes in the main address space and in the USER page. The USER page doesn't support write page, if the AP mode is selected writes in the USER page will be done in AQW mode. This avoids to change WMODE by software while mixing writes in the main address space and in the USER page.

**Procedure for Manual Page Writes (WMODE = MAN)**

The block to be written must be erased before the write command is given.

- Write to the page buffer by addressing the NVM main address space directly.
- Issue a Read-Memory-Barrier from the NVMCTRL AHB0 interface. A single read to AHB0 after writing the whole page buffer.
- Write the page buffer to memory:
  - CMD = WP (and CMDEX) to write the full content of the page buffer into the NVM at the page pointed by ADDR.
  - CMD = WQW (and CMDEX) to write into the NVM the page buffer quad word pointed by ADDR.
- The READY bit in the STATUS register will be low while programming is in progress, and access through the AHB in the same bank will be stalled.

**Procedure for Automatic Writes (WMODE = ADW or AQW or APW)**

The block to be written must be erased before the last write to the page buffer is performed. The internal write operation will begin when the second word is written for WMODE = ADW, when the fourth word is written for WMODE = AQW, and when the last word of the page is written for WMODE = APW.

Note that partially written pages must be written with a manual write.

If the command interface is already processing a command, the AHB is stalled until the automatic write command is taken. Therefore it is possible to chain write commands without polling STATUS.READY. For applications that must not stall the AHB bus the automatic write must be used carefully: STATUS.READY must be checked after each double-word or quad-word or page buffer write depending on WMODE before chaining with a new write to avoid stalling the bus.

- Write to the page buffer by addressing the NVM main address space directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - When the word location in the page buffer is written, the double word or quad word or page is automatically written to NVM main address space.
- STATUS.READY will be zero while programming is in progress and access through the AHB will be stalled.

**NVM Write Example (Manual Write Mode)**

1. Configure manual write for the NVM using WMODE (NVMCTRL.CTRLA).
2. Make sure the NVM is ready to accept a new command (NVMCTRL.STATUS).
3. Clear page buffer ( NVMCTRL.CTRLB).
4. Make sure NVM is ready to accept a new command (NVMCTRL.STATUS).
5. Clear the DONE Flag (NVMCTRL.INTFLAG).
6. Write data to page buffer with 32-bit accesses at the needed address.
7. Perform page write (NVMCTRL.CTRLB).
8. Make sure NVM is ready to accept a new command (NVMCTRL.STATUS).
9. Clear the DONE Flag (NVMCTRL.INTFLAG).

**Note:** The Flash region where the page belongs, must be unlocked in order to write it.

#### 25.6.6.3. Read While Write (RWW)

This feature makes it possible to program and read the NVM simultaneously without stalling the AHB bus independantly from any cache consideration. The basic principle is that NVM is made of two banks, one can be read while the other is programmed.

Limitations:

- It is not possible to read both banks simultaneously, reads will be prioritized and issued in series.
- It is not possible to program or erase both banks simultaneously, a new command will be accepted only after the completion of the previous one, otherwise the new command is ignored and INTFLAG.PROGE is set.
- RWW is not possible when reading or programming auxilliary pages, any read will result in an AHB stall and the command interface doesn't accept any command until completion of the previous one.

#### 25.6.6.4. Suspend/Resume

This feature can be enabled by writing a '1' to CTRLA.SUSPEN. Any modify operation, such as write/erase can be suspended even those triggered by the SmartEEPROM.

When enabled, the following commands are suspended by a NVM read request:

- EB
- WP

If a read occurs while executing one of the commands listed above, the NVMCTRL will follow these steps:

1. Send a suspend command to the NVM.
2. Wait for the NVM to be ready.
3. Read the NVM. The NVMCTRL will persist in this step when a new read request occurs or else proceed.
4. Resume the suspended operation.

A suspend operation will set INTFLAG.SUSP. To clear it, write a '1' to INTFLAG.SUSP.

The NVM suspended state is reflected in STATUS.SUSP.

Limitations:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Suspend is not possible for a read in the page being programmed.
- Suspend is not possible for a read in a sector (128 KB) containing a block under erase.
- It is not possible to enter power reduction mode when a command is suspended.

#### 25.6.6.5. Page Buffer

The page buffer is automatically cleared to *all-ones* after any page write operation (WP or WQW command). If a partial page has been written and it is desired to clear the contents of the page buffer, the Page Buffer Clear (PBC) command can be used. The status of the page buffer is given by STATUS.LOAD. This bit indicates that the NVM page buffer has been loaded with one or more words. Immediately after an NVM load has been performed, this flag is set, and it remains set until a WP or WQW or a PBC command is given.

The Page Buffer cannot be written while a write command is executing in the NVM. Trying to do so stalls the AHB bus. To avoid stalling the AHB bus, STATUS.READY can by polled prior to issue a write command.

Clearing the page buffer also clears to all ones the PBLDATA0 and PBLDATA1.

#### 25.6.6.6. Erase

Before a page can be written, it must be erased. The erase granularity depend on the address space (block or page). The Erase Block/Page command can be used to erase the desired block or page in the NVM main address space. Erasing the block/page sets all bits to '1'. If the block/page resides in a region that is locked, the erase will not be performed and the Lock Error bit in the INTFLAG register (INTFLAG.LOCKE) will be set. The INTFLAG.PROGE will also be set because the command did not complete. The Erase Page command can be issued on the USER page in the auxiliary space.

The procedure for an Erase Block/Page command is as follows:

1. Write the address of the block/page to erase to ADDR. Any address within the block/page can be used.
2. Issue an Erase Block/Page command.

The page buffer can be written while an erase page or erase block is being performed.

#### 25.6.6.7. Lock and Unlock Region

The commands LR and UR are used to lock and unlock regions. These commands only update the RUNLOCK register but not the corresponding field in the user page.

**Related Links**

Region Lock Bits

CTRLB

Control B

#### 25.6.6.8. Power Reduction Mode

The NVM implements a power reduction mode which cuts its static power consumption. If a command or a AHB access is issued the NVM is woken-up. The AHB access or the command are processed after the NVM wake-up time. The wake-up time can be reduced by enabling the PM fast wake-up feature, this is configured through the PM STDBYCFG register.

The NVM power reduction mode is entered depending on the CTRLA.PRM mode:

- MANUAL:
  - Power reduction mode entering conditions:
    - SPRM command
  - Power reduction mode leaving conditions:
    - AHB access (read or write)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

        - CPRM or any other command
- SEMIAUTO:
  - Power reduction mode entering conditions:
    - SPRM command
    - System enters Standby mode
    - AHB access completes while in Standby mode
    - Any command completes while in Standby mode
  - Power reduction mode leaving conditions:
    - AHB access (read or write)
    - CPRM or any other command
- FULLAUTO:
  - Power reduction mode entering conditions:
    - SPRM command
    - System enters Standby mode
    - AHB access completes while in Standby mode
    - Any command completes while in Standby mode
  - Power reduction mode leaving conditions:
    - AHB access (read or write)
    - CPRM or any other command
    - When the system leaves the Standby mode

STATUS.READY is high when NVM is in power reduction mode indicating that the module can accept a command.

STATUS.PRM is high when NVM is in power reduction mode.

**Note:** It is not possible to enter power reduction mode when a command is suspended. Automatic power reduction entry is postponed until the command resumes and completes. The SPRM command is discarded when STATUS.SUSP is high and INTFLAG.PROGE is set.

**Related Links**

PM – Power Manager

STDBYCFG

Standby Configuration

### 25.6.7. Safe Flash Update Using Dual Banks

This feature enables a firmware to execute from the NVM and simultaneously program the Flash with a new version of itself.

The new firmware has to be programmed in BANKB if STATUS.AFIRST = 1 or BANKA otherwise.

After programming is completed one can issue the BKSWRST command to swap the banks and to reset the device. The information of which BANK is mapped to the NVM main address space base address is self contained in the NVM using a special fuse that can be programmed or erased individually. This fuse is managed by the BKSWRST command. STATUS.AFIRST reflects the status of this fuse after Reset. The BKSWRST command is atomic meaning that no fetch in the NVM can occur while executing this command. This command executes with the following steps:

1. Stall AHB interfaces.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

2. If PARAM.SEE is '1' and 0 < SEESTAT.SBLK < 11, the NVMCTRL starts to reallocate the SmartEEPROM data to the first bank. Active SEES remains the same at the end of the reallocation.
3. Is STATUS.AFIRST = 1: program the AFIRST fuse (new value = 0) otherwise erase it (new value = 1)
4. Resets the device, after reset, RSTC RCAUSE indicates that the reset was triggered by the NVMCTRL.

After Reset the new firmware is executed from the last programmed bank.

If the SmartEEPROM is configured, the size of the the reserved space in Flash must not exceed the bank size. In other words 2*SEESTAT.SBLK.8192 must be lower than half the NVM size in Bytes. In situations where both the banks contain separate applications (or an application in one bank and a bootloader in the other bank), both the banks must have Flash area reserved for SmartEEPROM. This means that the usable area for code in each bank is 'Size of the Bank' minus the size of the Flash configured for the SmartEEPROM using SBLK Fuse.

## 25.6.8. SmartEEPROM

### 25.6.8.1. Principle of Operation

The SmartEEPROM feature is provided through the AHB2 interface and makes a portion of the NVM appear like a RAM. 8-bit, 16-bit, 32-bit access is supported.

The SmartEEPROM concept relies on the following NVM physical property: It is always possible to write a '0' in a NVM word, even if this word has been previously programmed - but it is not possible to write a '1' to a bit already programmed (holding a '0').

The algorithm consists of virtually mapping physical portions of the NVM to logical addresses with an indirection mechanism. A physical page is assigned to a virtual page address and is kept as long as no bit has to be flipped from '0' to '1', as this operation requires a full block erase. In case such a transition is required, a new physical page is assigned to the modified virtual page (placed in the Flash area reserved for the SmartEEPROM). Writing the virtual page affects the cycling endurance of the SmartEEPROM.

A region can overlap the SmartEEPROM region (depending on the allocated space for the SmartEEPROM), but SmartEEPROM is independent of the Region Lock Bits.

If NVMCTRL.STATUS.AFIRST contains '1', BANKA is mapped to the NVM main address space base address (0x0000). In this case, SmartEEPROM will be in BANKB. Conversely, when BANKB is mapped to the NVM main address space base address, SmartEEPROM will be in BANKA. Thus, the CPU is not halted when accessing the SmartEEPROM.

### 25.6.8.2. Address Spaces

The SmartEEPROM address space is divided into two distinct areas:

- DATA
  - Starts at offset 0x0
  - Size is 512B, 1 KB, 2 KB, 4 KB, 8 KB, 16 KB, 32 KB, 64 KB depending on SEESTAT.PSZ and SEESTAT.SBLK (refer to the "SmartEEPROM virtual size")
  - This area is write protected if SEESTAT.LOCK is set. SEESTAT.LOCK is non volatile.
  - The LSEE and USEE commands lock and unlock the SmartEEPROM.
- REGISTER:
  - Starts at offset 0x10000
  - Size is 20B
  - This area is write protected if either
    - SEESTAT.LOCK is set (non-volatile)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- SEESTAT.RLOCK is set (volatile).

The LSEER and USEER commands lock and unlock the SmartEEPROM register address space.

As a consequence SEESTAT.LOCK and SEESTAT.RLOCK must be low to write the SmartEEPROM register address space.

**Related Links**

SmartEEPROM Virtual Size

#### 25.6.8.3. Data Structures

The SmartEEPROM algorithm relies on two virtual sectors (SEES) physically located in the last blocks of the following:

- BANKB if STATUS.AFIRST = 1
- BANKA if STATUS.AFIRST = 0

Only one SEES is active at a time, the other must be erased, ready for data reallocation.

The current active SEES is indicated in SEESTAT.ASEES:

- 0: SEES0 is active
- 1: SEES1 is active

SEESTAT.ASEES is loaded after Reset from a special fuse in the NVM which can be programmed or erased individually. This fuse can be set by issuing the ASEES1 command or cleared by issuing the ASEES0 command. SEESTAT.ASEES reflects this change immediately.

The maximum number of virtual pages is limited to 128.

A page allocation consists of assigning a SEEP to a virtual page for the first time. A page reallocation consists of assigning a new SEEP to an already existing virtual page. In both cases the selected virtual page index and the next available page are written.

The SEEP size (PSZ) is configurable. The number of blocks allocated per SEES is configurable.

#### 25.6.8.4. SmartEEPROM Virtual Size

The SmartEEPROM interface virtual size is the maximum amount of data that can be stored in it. This defines the maximum size of this interface. Trying to read or write outside the boundaries throws an hardfault exception.

The SBLK bits indicate the number of blocks allocated per SmartEEPROM virtual sector. The SmartEEPROM raw data resides in the upper blocks of the NVM main address space but is not accessible through AHB0 nor AHB1. The SmartEEPROM interface maximum size depends on SEESTAT.PSZ and SEESTAT.SBLK:

**Table 25-6.** SmartEEPROM Virtual Size in Bytes

| SEESTAT.SBLK | SEESTAT.PSZ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | **512** | **1024** | **2048** | **4096** | 4096 | 4096 | 4096 | 4096 |
| 2 | *512* | *1024* | *2048* | *4096* | **8192** | 8192 | 8192 | 8192 |
| 3 | *512* | *1024* | *2048* | *4096* | *8192* | **16384** | 16384 | 16384 |
| 4 | *512* | *1024* | *2048* | *4096* | *8192* | *16384* | 16384 | 16384 |
| 5 | *512* | *1024* | *2048* | *4096* | *8192* | *16384* | **32768** | 32768 |
| 6 | *512* | *1024* | *2048* | *4096* | *8192* | *16384* | *32768* | 32768 |
| 7 | *512* | *1024* | *2048* | *4096* | *8192* | *16384* | *32768* | 32768 |
| 8 | *512* | *1024* | *2048* | *4096* | *8192* | *16384* | *32768* | 32768 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 25-6.** SmartEEPROM Virtual Size in Bytes (continued)

| SEESTAT.SBLK | SEESTAT.PSZ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
| 9 | *512* | *1024* | *2048* | *4096* | *8192* | *16384* | *32768* | 65536 |
| 10 | *512* | *1024* | *2048* | *4096* | *8192* | *16384* | *32768* | **65536** |

- The *italic* cells indicate sub-optimal configurations, unnecessary blocks are allocated.
- The **bold** cells indicate optimal valid configurations with the maximum number of SEEP depending on SEESTAT.PSZ and SEESTAT.SBLK .

#### 25.6.8.5. SmartEEPROM wear leveling

The wear leveling factor is the minimum ratio per which the access frequency to a physical flash cell is divided when the maximum number of SEEP in a SEES is reached. This maximum number is depends on the SEESTAT.PSZ and SEESTAT.SBLK user configuration. As there are two SEES the wear leveling is two times the maximum SEEP number.

**Table 25-7.** Wear leveling depending on SEESTAT.PSZ and SEESTAT.SBLK

| SEESTAT.PSZ: SEESTAT.SBLK | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|---|---|
| 0 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| 1 | 288 | 288 | 288 | 288 | **190** | **94** | **46** | **22** |
| 2 | 288 | 288 | 288 | 288 | 288 | **222** | **110** | **54** |
| 3 | 288 | 288 | 288 | 288 | 288 | 288 | **174** | **86** |
| 4 | 288 | 288 | 288 | 288 | 288 | 288 | **238** | **118** |
| 5 | 288 | 288 | 288 | 288 | 288 | 288 | 238 | **150** |
| 6 | 288 | 288 | 288 | 288 | 288 | 288 | 238 | **182** |
| 7 | 288 | 288 | 288 | 288 | 288 | 288 | 238 | **214** |
| 8 | 288 | 288 | 288 | 288 | 288 | 288 | 238 | **246** |
| 9 | 288 | 288 | 288 | 288 | 288 | 288 | 288 | **278** |
| 10 | 288 | 288 | 288 | 288 | 288 | 288 | 288 | 288 |

#### 25.6.8.6. Writing and Reading the SmartEEPROM

SEESTAT.LOCK must be '0'; otherwise, writes are discarded and a hardfault exception is thrown. SmartEEPROM write access can be locked with the LSEE command and unlocked with the USEE command.

1. Configure SBLK and PSZ fuses to define the SmartEEPROM total size and size of each page.
2. Define a pointer to the SmartEEPROM area. It can be used for 8-, 16- or 32-bit access.

```
volatile uint8_t *SmartEEPROM8 = (uint8_t *) SMEEPROM_ADDR; volatile uint16_t
*SmartEEPROM16 = (uint16_t *) SMEEPROM_ADDR; volatile uint32_t *SmartEEPROM32 = (uint32_t
*) SMEEPROM_ADDR;
```

3. Wait until SmartEEPROM is busy.

```
while (NVMCTRL->SEESTAT.bit.BUSY);
```

4. Write to the SmartEEPROM like writing a RAM location. Perform an 8-, 16- or 32-bit write.
5. If automatic reallocation is disabled with SEECFG.APRDIS, check the SEESFULL interrupt flag to ensure that the active SmartEEPROM sector is not full.
6. To read back the content, read the location using the defined pointer.

```
uint8_t eep_data_8 = 0; while (NVMCTRL->SEESTAT.bit.BUSY); eep_data_8 = SmartEEPROM8[0];
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

There are two NVM pagebuffer management modes available, selected by writing the SEECFG.WMODE bit field:

- UNBUFFERED (default): WP command triggered after any pagebuffer update
- BUFFERED: WP command triggered only in case of NVM page crossing. This mode increases the NVM wear-leveling but is more sensitive to power loss. SEESTAT.LOAD is high when the pagebuffer contains unwritten data.

When SEECFG.WMODE selects the buffered mode, the page buffer can contain unwritten SmartEEPROM data. This is reflected by SEESTAT.LOAD. To flush the SmartEEPROM data inside the page buffer, issue the SEEFLUSH command.

INTFLAG.SEEWRC indicates when a AHB write to the SmartEEPROM has completed:

1. Unbuffered mode: AHB write has completed, NVM is programmed with correct values except if INTFLAG.SEESOVF was thrown.
2. Buffered mode: AHB write has completed,
   - if SEESTAT.LOAD = 0: NVM is programmed with correct values, except if INTFLAG.SEESOVF was thrown.
   - otherwise; new data is in the page buffer, but is not yet programmed in the NVM.

### 25.6.8.7. SmartEEPROM Sector Reallocation

By default, the SEES reallocation is performed in hardware when the next available page in the Host index reaches the maximum SEEP number. Automatic reallocation can be disabled by writing a one in SEECFG.APRDIS. The sector reallocation can also be triggered manually by issuing the SEERALOC command. The SEES reallocation process consists of the following:

- Erase the non-active sector.
- Copying valid data from the active sector to another sector, old data is filtered.
- Swap ASEES either by issuing the ASEES1 command if SEESTAT.ASEES is reading '0' or by issuing the ASEES0 command if SEESTAT.ASEES is read as '1'.

By default, this process is automatically handled by hardware, and indicated by the SEESTAT.BUSY flag. If in buffered mode, the page buffer must be flushed before triggering a reallocation; otherwise, the content of the pagebuffer will be lost.

**Notes:**

1. The BKSWRST command triggers automatically the reallocation algorithm which operates as described above except copy is done in the same active sector but in the first bank. This operation is atomic, meaning that no modify operation can be issued in the mean time.
2. At the end of reallocation, there may be a CPU stall for a few milliseconds. Critical code can be placed in TCM, RAM, or CMCC.

As the total size of the whole SEEP exceeds the SmartEEPROM virtual size for a given configuration, there is always free SEEP to replace existing data. In the case all addresses have been written, after sector reallocation the number of free SEEP is given in the following table.

**Table 25-8.** Minimum number of free SEEP after sector reallocation

| SEESTAT.PSZ: SEESTAT.SBLK | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|---|---|
| 0 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| 1 | 16 | 16 | 16 | 16 | 31 | 15 | 7 | 3 |
| 2 | 16 | 16 | 16 | 16 | 16 | 47 | 23 | 11 |
| 3 | 16 | 16 | 16 | 16 | 16 | 16 | 23 | 11 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 25-8.** Minimum number of free SEEP after sector reallocation (continued)

| SEESTAT.PSZ: SEESTAT.SBLK | 4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 |
|---|---|---|---|---|---|---|---|---|
| 4 | 16 | 16 | 16 | 16 | 16 | 16 | 55 | 27 |
| 5 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 11 |
| 6 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 27 |
| 7 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 43 |
| 8 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 59 |
| 9 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 11 |
| 10 | 16 | 16 | 16 | 16 | 16 | 16 | 16 | 16 |

### 25.6.9. NVM User Configuration

The NVM user configuration resides in the auxiliary space. Refer to the *Physical Memory Map and Product Mapping* of the device for calibration and auxiliary space address mapping.

The NVM user configuration is as follows:

- The bootloader size. The bootloader resides in the main array starting at offset zero. The allocated bootloader section is protected against erase or write operations including the chip erase operation.
- The SmartEEPROM number of blocks per SEES (SBLK bits). This configuration is loaded after a reset into the SEESTAT.SBLK bits.
- The SmartEEPROM virtual page size (PSZ bits). This configuration is loaded after a reset into the SEESTAT.PSZ bits.
- The region lock bits (reflected in the RUNLOCK register).
- The SmartEEPROM RUNLOCK bit (reflected in SEESTAT.LOCK).

**Table 25-9.** Boot Loader Size

| BOOTPROT [3:0] | Rows Protected by BOOTPROT | Boot Loader Size in KBytes |
|---|---|---|
| 15 | None | 0 |
| 14 | 1 | 8 |
| 13 | 2 | 16 |
| 12 | 3 | 24 |
| 11 | 4 | 32 |
| 10 | 5 | 40 |
| 9 | 6 | 48 |
| 8 | 7 | 56 |
| 7 | 8 | 64 |
| 6 | 9 | 72 |
| 5 | 10 | 80 |
| 4 | 11 | 88 |
| 3 | 12 | 96 |
| 2 | 13 | 104 |
| 1 | 14 | 112 |
| 0 | 15 | 120 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 25-10.** SmartEEPROM Allocated Space

| SBLK[4:0] | Total Blocks | Bytes |
|---|---|---|
| 10 | 20 | 163840 |
| 9 | 18 | 147456 |
| 8 | 16 | 131072 |
| 7 | 14 | 114688 |
| 6 | 12 | 98304 |
| 5 | 10 | 81920 |
| 4 | 8 | 65536 |
| 3 | 6 | 49152 |
| 2 | 4 | 32768 |
| 1 | 2 | 16384 |
| 0 | 0 | 0 |

**Table 25-11.** SmartEEPROM Virtual Page Size

| PSZ[2:0] | Page Size in Bytes |
|---|---|
| 7 | 512 |
| 6 | 256 |
| 5 | 128 |
| 4 | 64 |
| 3 | 32 |
| 2 | 16 |
| 1 | 8 |
| 0 | 4 |

**Related Links**
Physical Memory Map
Product Memory Mapping Overview

### 25.6.10. Security Bit

The security bit allows the entire chip to be locked from external access for code security.

**Related Links**
DSU - Device Service Unit

#### 25.6.10.1. Security Bit Set Procedure

The Security Bit Set procedure is as follows:

1. Issue the Set Security Bit (SSB) command. If a debugger is connected, it still has access to the device after issuing this command (DSU STATUSB.PROT = 0).
2. Check NVMNCTRL.INTFLAG.PROGE and NVMNCTRL.INTFLAG.DONE.
3. Reset the NVMCTRL peripheral or the device.
4. After reset, DSU STATUSB.PROT = 1.

To reflect the NVM security bit state correctly, the NVMCTRL needs to replay the start-up procedure. This is done by issuing a SWRST command or by resetting the device.

**Related Links**
DSU - Device Service Unit


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.6.10.2. Security Bit Clear Procedure

The only way to clear the security bit is through a debugger Chip Erase command. The NVM security bit is cleared after all internal volatile and NVM have been cleared. The device protection status is updated at the end of the command meaning that no reset is necessary.

**Related Links**

DSU - Device Service Unit

### 25.6.11. Line Cache

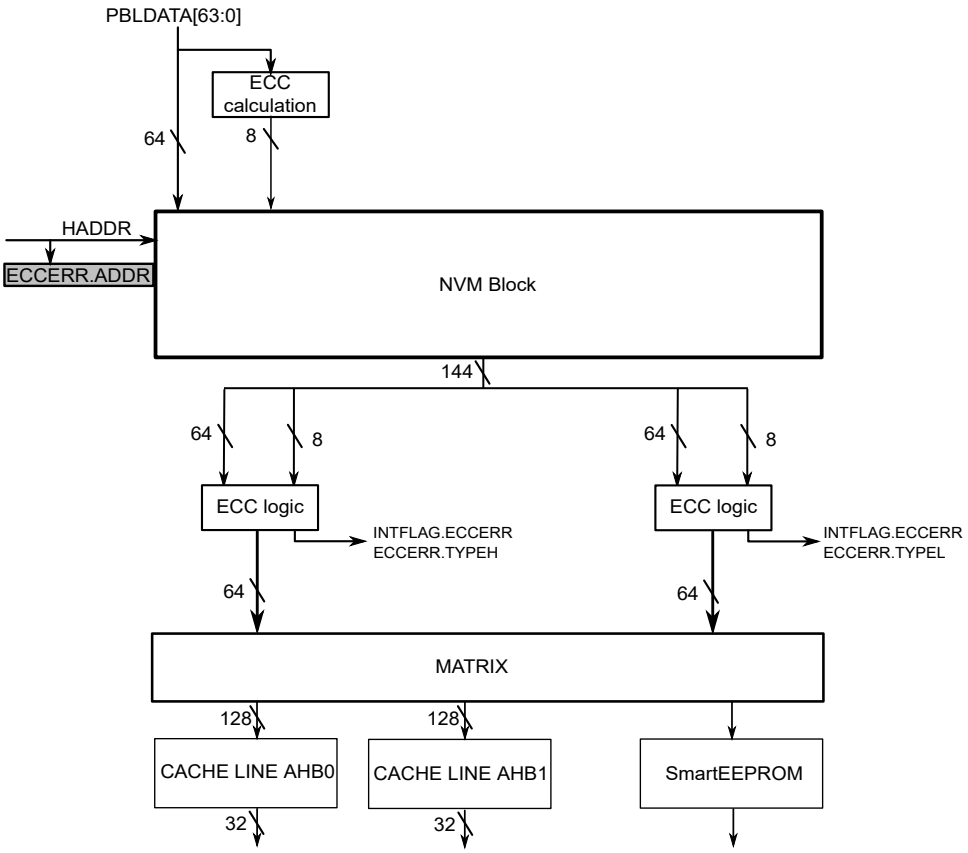
NVM reads 128-bit at a time. AHB0 and AHB1 interfaces implement each a 128-bit cache line.This reduces the device power consumption when reading continuous data and improves system performance when wait states are required. Line cache are enabled by default and can be individually disabled per AHB interface by writing a one in the CACHEDIS[0] or CACHEDIS[1] bit in the CTRLA register (CTRLA.CACHEDIS[1:0]). Refer to CTRLA register description for more details. Commands affecting NVM content automatically invalidate cache lines.

### 25.6.12. Error Correction Code (ECC)

Error Correcting Code (ECC) is implemented to detect and correct errors that may arise in the NVM array. ECC is by default enabled and cannot be disabled by the user.

### 25.6.12.1. Block Diagram

**Figure 25-2.** ECC Diagram




Note that the ECC correction is disabled when access is performed by the SmartEEPROM interface.

### 25.6.12.2. ECC Error Detection

The NVM physical block fetches 128-bit quad-word and ECC checking is performed on a 64-bit basis independently on the low and high double-words. Therefore two ECC decoders operate in parallel.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

An ECC failure may be present in any of the four words from the NVM, not necessarily the word that is addressed on the bus. Any ECC error in a double-word will be reported the first time the quad-word access. The ECC logic in the read data path is capable of double error detection and single error correction on the fly per 64-bit double-word.

Upon detection:

- INTFLAG ECC error flags are updated:
  - The ECC single error interrupt flag is raised (INTFLAG.ECCSE) in case of single error
  - The ECC dual error interrupt flag is raised (INTFLAG.ECCDE) in case of dual error
- ECCERR.ADDR is updated with the faulty quad-word byte address in the main address space.
- ECCERR.TYPEL is updated with the error type (NONE, SINGLE, DUAL) detected on the low 64-bit double word.
- ECCERR.TYPEH is updated with the error type (NONE, SINGLE, DUAL) detected on the high 64-bit double word.

INTFLAG.ECCSE and INTFLAG.ECCDE are automatically cleared when ECCERR is read.

ECCERR.TYPEL and ECCERR.TYPEH are reset to the NONE value when ECCERR is read. If an error occurs while reading ECCERR, the previous error information is sent to the APB and ECCERR is updated with the next error information.

If a single-error has been detected and INTFLAG.ECCSE or INTFLAG.ECCDE is not clear:

- Any incoming single-errors is ignored
- First incoming dual-error overrides ECCERR.ADDR, ECCERR.TYPEL and ECCERR.TYPEH

If a dual-error has been detected and INTFLAG.ECCDE is not clear:

- incoming single-errors are ignored
- incoming dual-errors are ignored

ECCERR.ADDR is always quad-word aligned. If jumping to a word that is not quad-word aligned, e.g. jumping to address 0x100C, INTFLAG.ECCDE and INTFLAG.ECCSE are updated according to the types of detected errors, and ECCERR.ADDR will read 0x1000, irrespective of whether the ECC error was in address 0x1000, 0x1004, 0x1008, or 0x100C.

### 25.6.13. Reset During Operation

Program or erase operations must not be interrupted. The content of a block or a page is unpredictable in case of reset during either an erase or a write operation. To reduce the risk of having a BOD reset due to a power loss one can monitor the external voltage before issuing any program or erase operation. The user can also prefer the WQW command instead of the WP command as a short command is more likely to complete successfully than a long one with a given external decoupling capacitor. In case of reset during a write or erase operation the impacted block must be erased before being read or programmed as its content is unknown.

### 25.6.14. Chip Erase

The chip erase operation is system-wide and issued through the DSU.

The chip erase procedure is as follows:

1. Volatile memories are cleared and NVM array is erased simultaneously (except the BOOTPROT section).
2. Special individual fuses are set as follows:
   - If the BOOTPROT section is not defined, then NVMCTRL STATUS.AFIRST = 1 otherwise it is left unchanged.
   - NVMCTRL SEESTAT.ASEES = 1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- NVMCTRL SEESTAT.LOCK = 0.
- DSU STATUSB.CELCK = 0.

3. The Security bit is cleared provided no internal error has been detected in the previous steps:
   - If all internal NVM verify operations have succeeded: go to 4.
   - Otherwise, set DSU STATUSA.DONE and DSU STATUSA.FAIL and exit.
4. DSU STATUSB.PROT is cleared, the system is not protected.

**Note:** The CB, FS, USER pages (in the auxiliary address space) and the section allocated as a bootloader using BOOTPROT are not affected by the chip erase operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 25.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | PRM[1:0] | | WMODE[1:0] | | SUSPEN | AUTOWS | | |
| | | 15:8 | CACHEDIS1 | CACHEDIS0 | AHBNS1 | AHBNS0 | RWS[3:0] | | | |
| 0x02 ... 0x03 | Reserved | | | | | | | | | |
| 0x04 | CTRLB | 7:0 | | CMD[6:0] | | | | | | |
| | | 15:8 | CMDEX[7:0] | | | | | | | |
| 0x06 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | PARAM | 7:0 | NVMP[7:0] | | | | | | | |
| | | 15:8 | NVMP[15:8] | | | | | | | |
| | | 23:16 | | | | | | PSZ[2:0] | | |
| | | 31:24 | SEE | | | | | | | |
| 0x0C | INTENCLR | 7:0 | SUSP | NVME | ECCDE | ECCSE | LOCKE | PROGE | ADDRE | DONE |
| | | 15:8 | | | | | | SEEWRC | SEESOVF | SEESFULL |
| 0x0E | INTENSET | 7:0 | SUSP | NVME | ECCDE | ECCSE | LOCKE | PROGE | ADDRE | DONE |
| | | 15:8 | | | | | | SEEWRC | SEESOVF | SEESFULL |
| 0x10 | INTFLAG | 7:0 | SUSP | NVME | ECCDE | ECCSE | LOCKE | PROGE | ADDRE | DONE |
| | | 15:8 | | | | | | SEEWRC | SEESOVF | SEESFULL |
| 0x12 | STATUS | 7:0 | | | BPDIS | AFIRST | SUSP | LOAD | PRM | READY |
| | | 15:8 | | | | | BOOTPROT[3:0] | | | |
| 0x14 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | RUNLOCK | 7:0 | RUNLOCK[7:0] | | | | | | | |
| | | 15:8 | RUNLOCK[15:8] | | | | | | | |
| | | 23:16 | RUNLOCK[23:16] | | | | | | | |
| | | 31:24 | RUNLOCK[31:24] | | | | | | | |
| 0x1C | PBLDATA0 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x20 | PBLDATA1 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x24 | ECCERR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | TYPEH[1:0] | | TYPEL[1:0] | | | | | |
| 0x28 | DBGCTRL | 7:0 | | | | | | | ECCELOG | ECCDIS |
| 0x29 | Reserved | | | | | | | | | |
| 0x2A | SEECFG | 7:0 | | | | | | | APRDIS | WMODE |
| 0x2B | Reserved | | | | | | | | | |
| 0x2C | SEESTAT | 7:0 | | | | RLOCK | LOCK | BUSY | LOAD | ASEES |
| | | 15:8 | | | | | SBLK[3:0] | | | |
| | | 23:16 | | | | | | PSZ[2:0] | | |
| | | 31:24 | | | | | | | | |

## 25.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 25.8.1. Control A

**Name:** CTRLA
**Offset:** 0x0
**Reset:** 0x0004
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CACHEDIS1 | CACHEDIS0 | AHBNS1 | AHBNS0 | RWS[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PRM[1:0] | | WMODE[1:0] | | SUSPEN | AUTOWS | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | | |

**Bit 15 – CACHEDIS1** AHB1 Cache Disable
AHB1 interface cache disable.
0: cache line is enabled
1: cache line is disabled
Cache lines are automatically invalidated when a write or erase operation is started in the NVM.

**Bit 14 – CACHEDIS0** AHB0 Cache Disable
AHB0 interface cache disable.
0: cache line is enabled
1: cache line is disabled
Cache lines are automatically invalidated when a write or erase operation is started in the NVM.

**Bit 13 – AHBNS1** Force AHB1 access to Non-Sequential
This bit forces AHB1 communication to be non-sequential.

| Value | Description |
|---|---|
| 0 | AHB sequential accesses remain sequential. |
| 1 | AHB sequential accesses are forced to non-sequential, therefore forcing rearbitration for each access. |

**Bit 12 – AHBNS0** Force AHB0 access to Non-Sequential
This bit forces AHB0 communication to be non-sequential.

| Value | Description |
|---|---|
| 0 | AHB sequential accesses remain sequential. |
| 1 | AHB sequential accesses are forced to non-sequential, therefore forcing rearbitration for each access. |

**Bits 11:8 – RWS[3:0]** NVM Read Wait States
These bits give the number of wait states for a read operation when AUTOWS=0. Zero indicates zero wait states, one indicates one wait state, etc., up to 15 wait states.
This register is initialized to 0 wait states. Software can change this value based on the NVM access time and system frequency.

**Bits 7:6 – PRM[1:0]** Power Reduction Mode during Sleep
Indicates the power reduction mode during sleep.

| Value | Name | Description |
|---|---|---|
| 0x0 | SEMIAUTO | NVM block enters low-power mode when entering standby mode. NVM block enters low-power mode when SPRM command is issued. NVM block exits low-power mode upon first access. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x1 | FULLAUTO | NVM block enters low-power mode when entering standby mode. NVM block enters low-power mode when SPRM command is issued. NVM block exits low-power mode when system is not in standby mode. |
| 0x2 | | Reserved |
| 0x3 | MANUAL | NVM block does not enter low-power mode when entering standby mode. NVM block enters low-power mode when SPRM command is issued. NVM block exits low-power mode upon first access. |

**Bits 5:4 – WMODE[1:0]** Write Mode

Write commands can be generated automatically when crossing address boundaries while writing to the NVM. Boundaries depend on the settings below.

| Value | Name | Description |
|---|---|---|
| 0x0 | MAN | Manual Write |
| 0x1 | ADW | Automatic Double Word Write |
| 0x2 | AQW | Automatic Quad Word |
| 0x3 | AP | Automatic Page Write |

**Bit 3 – SUSPEN** Suspend Enable

0: The write and erase suspend resume feature is disabled.
1: A write or erase operation can be suspended in case of a read in the same bank.

**Bit 2 – AUTOWS** Auto Wait State Enable

0: Automatic wait state generation is disabled. The number of wait states used is given by CTRLA.RWS bits.
1: Automatic wait state generation is enabled. The number of wait states used is automatically detected therefore the module can operate at any frequency up to the device maximum frequency. A minimum of one cycle latency is induced.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CMDEX[7:0] | | | | | | | |
| Access | PAC Write-Protection | PAC Write-Protection | PAC Write-Protection | PAC Write-Protection | PAC Write-Protection | PAC Write-Protection | PAC Write-Protection | PAC Write-Protection |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | CMD[6:0] | | | | | | |
| Access | | W | W | W | W | W | W | W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:8 – CMDEX[7:0]** Command Execution

This bit group should be written with the key value 0xA5 to enable the command written to CMD to be executed. If the bit group is written with a different key value, the write is not performed and INTFLAG.PROGE is set. PROGE is also set if the a previously written command is not complete.
The key value must be written at the same time as CMD. If a command is issued through the APB bus on the same cycle as an AHB bus access, the AHB bus access will be given priority. The command will then be executed when the NVM block and the AHB bus are idle.
STATUS.READY must be one when the command is issued.
INTFLAG.DONE is set when the command completes.

| Value | Name | Description |
|---|---|---|
| 0xA5 | KEY | Execution Key |
| Other | - | Reserved |

**Bits 6:0 – CMD[6:0]** Command

These bits define the command to be executed when the CMDEX key is written.

| Value | Name | Description |
|---|---|---|
| 0x0 | EP | Erase Page - Only supported in the User page in the auxiliary space. |
| 0x1 | EB | Erase Block - Erases the block addressed by the ADDR register, not supported in the user page |
| 0x2 | | Reserved |
| 0x3 | WP | Write Page - Writes the contents of the page buffer to the page addressed by the ADDR register, not supported in the user page |
| 0x4 | WQW | Write Quad Word - Writes a 128-bit word at the location addressed by the ADDR register. |
| 0x5-0xF | | Reserved |
| 0x10 | SWRST | Software Reset - Power-Cycle the NVM memory and replay the device automatic calibration procedure and resets the module configuration registers |
| 0x11 | LR | Lock Region - Locks the region containing the address location in the ADDR register until next reset. |
| 0x12 | UR | Unlock Region - Unlocks the region containing the address location in the ADDR register until next reset. |
| 0x13 | SPRM | Sets the power reduction mode. |
| 0x14 | CPRM | Clears the power reduction mode. |
| 0x15 | PBC | Page Buffer Clear - Clears the page buffer. |
| 0x16 | SSB | Set Security Bit |
| 0x17 | BKSWRST | Bank swap and system reset, if SmartEEPROM is used also reallocate its data into the opposite BANK |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x18 | CELCK | Chip Erase Lock - DSU CTRL.CE command is not available. As soon as the CELCK command is successfully executed, the chip erase capability is disabled and Microchip's failure analysis capabilities are limited. Therefore, the software has to ensure there is a way to unlock the chip erase by executing the CEULCK command. |
| 0x19 | CEULCK | Chip Erase Unlock - The DSU CTRL.CE command is available. |
| 0x1A | SBPDIS | Sets STATUS.BPDIS, bootloader protection is discarded until CBPDIS is issued or next start-up sequence. |
| 0x1B | CBPDIS | Clears STATUS.BPDIS, bootloader protection is not discarded. |
| 0x1C-0x2F | | Reserved |
| 0x30 | ASEES0 | Configure SmartEEPROM to use Sector 0 |
| 0x31 | ASEES1 | Configure SmartEEPROM to use Sector 1 |
| 0x32 | SEERALOC | Starts SmartEEPROM sector reallocation algorithm |
| 0x33 | SEEFLUSH | Flush SmartEEPROM data when in buffered mode |
| 0x34 | LSEE | Lock access to SmartEEPROM data from any means |
| 0x35 | USEE | Unlock access to SmartEEPROM data |
| 0x36 | LSEER | Lock access to the SmartEEPROM Register Address Space (above 64KB) |
| 0x37 | USEER | Unock access to the SmartEEPROM Register Address Space (above 64KB) |
| 0x38-0x7F | | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.3. NVM Parameter

**Name:** PARAM
**Offset:** 0x08
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SEE | | | | | | | |
| Access | R | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | PSZ[2:0] | | |
| Access | | | | | | R | R | R |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NVMP[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NVMP[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | | | | | | | | |

**Bit 31 – SEE** SmartEEPROM Supported
0: No SmartEEPROM support
1: SmartEEPROM is supported.

**Bits 18:16 – PSZ[2:0]** Page Size
Indicates the page size. Not all device families will provide all the page sizes indicated in the table.

| Value | Name | Description |
|---|---|---|
| 0x0 – 0x5 | Reserved | Reserved |
| 0x6 | 512 | 512 bytes |
| 0x7 | Reserved | Reserved |

**Bits 15:0 – NVMP[15:0]** NVM Pages
Indicates the number of pages in the NVM main address space


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SEEWRC | SEESOVF | SEESFULL |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SUSP | NVME | ECCDE | ECCSE | LOCKE | PROGE | ADDRE | DONE |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 10 – SEEWRC** SEE Write Completed Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the SEEWRC interrupt enable.
This bit will read as the current value of the SEEWRC interrupt enable.

**Bit 9 – SEESOVF** Active SEES Overflow Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the SEESOVF interrupt enable.
This bit will read as the current value of the SEESOVF interrupt enable.

**Bit 8 – SEESFULL** Active SEES Full Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the SEESFULL interrupt enable.
This bit will read as the current value of the SEESFULL interrupt enable.

**Bit 7 – SUSP** Suspended Write Or Erase Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the SUSP interrupt enable.
This bit will read as the current value of the SUSP interrupt enable.

**Bit 6 – NVME** NVM Error Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the NVME interrupt enable.
This bit will read as the current value of the NVME interrupt enable.

**Bit 5 – ECCDE** ECC Dual Error Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the ECCDE interrupt enable.
This bit will read as the current value of the ECCDE interrupt enable.

**Bit 4 – ECCSE** ECC Single Error Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the ECCSE interrupt enable.
This bit will read as the current value of the ECCSE interrupt enable.

**Bit 3 – LOCKE** Lock Error Interrupt Clear
Writing a zero to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit clears the LOCKE interrupt enable.
This bit will read as the current value of the LOCKE interrupt enable.

**Bit 2 – PROGE** Programming Error Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the PROGE interrupt enable.
This bit will read as the current value of the PROGE interrupt enable.

**Bit 1 – ADDRE** Address Error Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the ADDRE interrupt enable.
This bit will read as the current value of the ADDRE interrupt enable.

**Bit 0 – DONE** Command Done Interrupt Clear
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DONE interrupt enable.
This bit will read as the current value of the DONE interrupt enable.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 25.8.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x0E
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SEEWRC | SEESOVF | SEESFULL |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SUSP | NVME | ECCDE | ECCSE | LOCKE | PROGE | ADDRE | DONE |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 10 – SEEWRC** SEE Write Completed Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the SEEWRC interrupt enable.
This bit will read as the current value of the SEEWRC interrupt enable.

**Bit 9 – SEESOVF** Active SEES Overflow Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the SEESOVF interrupt enable.
This bit will read as the current value of the SEESOVF interrupt enable.

**Bit 8 – SEESFULL** Active SEES Full Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the SEESFULL interrupt enable.
This bit will read as the current value of the SEESFULL interrupt enable.

**Bit 7 – SUSP** Suspended Write Or Erase Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the SUSP interrupt enable.
This bit will read as the current value of the SUSP interrupt enable.

**Bit 6 – NVME** NVM Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the NVME interrupt enable.
This bit will read as the current value of the NVME interrupt enable.

**Bit 5 – ECCDE** ECC Dual Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the ECCDE interrupt enable.
This bit will read as the current value of the ECCDE interrupt enable.

**Bit 4 – ECCSE** ECC Single Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the ECCSE interrupt enable.
This bit will read as the current value of the ECCSE interrupt enable.

**Bit 3 – LOCKE** Lock Error Interrupt Enable
Writing a zero to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a one to this bit sets the LOCKE interrupt enable.
This bit will read as the current value of the LOCKE interrupt enable.

**Bit 2 – PROGE** Programming Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the PROGE interrupt enable.
This bit will read as the current value of the PROGE interrupt enable.

**Bit 1 – ADDRE** Address Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the ADDRE interrupt enable.
This bit will read as the current value of the ADDRE interrupt enable.

**Bit 0 – DONE** Command Done Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit sets the DONE interrupt enable.
This bit will read as the current value of the DONE interrupt enable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x10
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SEEWRC | SEESOVF | SEESFULL |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SUSP | NVME | ECCDE | ECCSE | LOCKE | PROGE | ADDRE | DONE |
| Access | R/W | R/W | R | R | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 10 – SEEWRC** SEE Write Completed

- Unbuffered mode:
  0: AHB write is pending.
  1: AHB write has completed, and NVM is programmed with correct values.
- Buffered mode:
  0: AHB write is pending.
  1: AHB write has completed.
  If SEESTAT.LOAD=0, then the NVM is programmed with correct values.
  If SEESTAT.LOAD=1, then data is still pending in the Page Buffer.

**Bit 9 – SEESOVF** Active SEES Overflow
0: No SEES overflow have been detected since the last clear.
1: At least SEES overflow has been detected since the last clear.
This bit can be cleared by writing a one to its bit location.

**Bit 8 – SEESFULL** Active SEES Full
0: The active SEES is not full
1: The active SEES is Full, meaning that the next write will fail if the active sector is not reallocated.
This bit can be cleared by writing a one to its bit location.

**Bit 7 – SUSP** Suspended Write Or Erase Operation
0: No write/suspend has occurred since the last clear.
1: A write or erase operation has been suspended since the last clear.
This bit can be cleared by writing a one to its bit location.

**Bit 6 – NVME** NVM Error
0: No NVM errors have been received since the last clear.
1: At least one NVM error has occurred since the last clear.
This bit can be cleared by writing a one to its bit location.

**Bit 5 – ECCDE** ECC Dual Error
0: No ECC dual errors have been received since the last ECCERR register read.
1: At least one ECC error has occurred since the last ECCERR register read.
This bit is cleared when the ECCERR register is read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 4 – ECCSE** ECC Single Error
0: No ECC single errors have been received since the last ECCERR register read.
1: At least one ECC error has occurred since the last ECCERR register read.
This bit is cleared when the ECCERR register is read.

**Bit 3 – LOCKE** Lock Error
0: No LOCK errors have been received since the last clear.
1: At least one LOCK error has occurred since the last clear.
This bit can be cleared by writing a one to its bit location.

**Bit 2 – PROGE** Programming Error
0: No PROG errors have been received since the last clear.
1: At least one PROG error has occurred since the last clear.
This bit can be cleared by writing a one to its bit location.

**Bit 1 – ADDRE** Address Error
0: No ADDRE error has been detected since the last clear.
1: At least one ADDRE error has been detected since the last clear.
This bit can be cleared by writing a one to its bit location.

**Bit 0 – DONE** Command Done
0: The NVM controller has not completed any command since the last clear.
1: At least one command has completed since the last clear.
This bit can be cleared by writing a one to its bit location.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.7. Status

**Name:** STATUS
**Offset:** 0x12
**Reset:** 0x0000
**Property:** Read-Only

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | BOOTPROT[3:0] | | | |
| Access | | | | | R | R | R | R |
| Reset | | | | | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | BPDIS | AFIRST | SUSP | LOAD | PRM | READY |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 1 | 0 | 0 | 0 | 0 |

**Bits 11:8 – BOOTPROT[3:0]** Boot Loader Protection Size
This bitfield is loaded from the NVM User page during the device startup (See the Memories chapter).
Defines the size of the BOOTPROT region which is protected against write or erase or Chip-Erase operations. This size is given by the following formula (15-BOOTPROT)*8KB.

**Bit 5 – BPDIS** Boot Loader Protection Disable
0: Boot loader protection is not discarded.
1: Boot loader protection against modify operations is discarded until CBPDIS is issued or next start-up sequence except for Chip-Erase.

**Bit 4 – AFIRST** BANKA First
0: Start address of bank B is mapped at 0x0000_0000.
1: Start address of bank A is mapped at 0x0000_0000.

**Bit 3 – SUSP** NVM Write Or Erase Operation Is Suspended
0: The NVM controller is not in suspended state.
1: The NVM controller is in suspended state.

**Bit 2 – LOAD** NVM Page Buffer Active Loading
This bit indicates that the NVM page buffer has been loaded with one or more words. Immediately after an NVM load has been performed, this flag is set, and it remains set until a Write Page (WP), Write Quad Word (WQW) or a page buffer clear (PBCLR) command is given.

**Bit 1 – PRM** Power Reduction Mode
This bit indicates the current NVM power reduction state. The NVM block can be set in power reduction mode in two ways: through the command interface or automatically when entering sleep with CTRLA.PRM set accordingly. PRM can be cleared in three ways: through AHB access to the NVM block, through the command interface (SPRM and CPRM) or when exiting sleep with CTRLA.PRM set accordingly.
0: NVM is not in power reduction mode
1: NVM is in power reduction mode.

**Bit 0 – READY** Ready to accept a command
0: The NVM controller is busy programming or erasing.
1: The NVM controller is ready to accept a new command.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.8. Address

**Name:** ADDR
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:0 – ADDR[23:0]** NVM Address

ADDR drives the hardware address to the NVM when a command is executed using CMDEX. It is a Byte address. This register is also automatically updated when writing to the page buffer or when writing the SmartEEPROM.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.9. Lock Section

**Name:** RUNLOCK
**Offset:** 0x18
**Reset:** 0xXXXXXXXX
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RUNLOCK[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RUNLOCK[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RUNLOCK[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNLOCK[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

**Bits 31:0 – RUNLOCK[31:0]** Region Un-Lock Bits
In order to set or clear these bits, the CMD register must be used to execute Lock Region and Unlock Region commands.
0: The corresponding region is locked.
1: The corresponding region is not locked.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.10. Page Buffer Load Data x

**Name:** PBLDATAn
**Offset:** 0x1C + n*0x04 [n=0..1]
**Reset:** 0xFFFFFFFF
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 31:0 – DATA[31:0]** Page Buffer Data


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.11. ECC Error Status

**Name:** ECCERR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** -

This register tracks errors on the NVM read path.

ECC error tracking is active until an error is detected. It is still active in case of single error but no dual error. In this case only a dual error can override this register status as a dual error is more critical than a single error. Error tracking resumes as soon as this register is read.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TYPEH[1:0] | | TYPEL[1:0] | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:30 – TYPEH[1:0]** High Double-Word Error Type
Indicates the type of error detected on the NVM 64-bit most significant read word. It is reset to None when this register is read except if an error occurs in the same cycle.

| Value | Name | Description |
|---|---|---|
| 0x0 | None | No Error Detected Since Last Read |
| 0x1 | Single | At Least One Single Error Detected Since last Read |
| 0x2 | Dual | At Least One Dual Error Detected Since Last Read |
| 0x3 | | Reserved |

**Bits 29:28 – TYPEL[1:0]** Low Double-Word Error Type
Indicates the type of error detected on the NVM 64-bit less significant read word. It is reset to None when this register is read except if an error occurs in the same cycle.

| Value | Name | Description |
|---|---|---|
| 0x0 | None | No Error Detected Since Last Read |
| 0x1 | Single | At Least One Single Error Detected Since last Read |
| 0x2 | Dual | At Least One Dual Error Detected Since Last Read |
| 0x3 | | Reserved |

**Bits 23:0 – ADDR[23:0]** Error Address
Indicates the Byte address of the last detected error.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.12. Debug Control

**Name:** DBGCTRL
**Offset:** 0x28
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ECCELOG | ECCDIS |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ECCELOG** Debugger ECC Error Tracking Mode
0: ECC errors detected during a read initiated by a debugger are not logged.
1: ECC errors detected during a read initiated by a debugger are logged.

**Bit 0 – ECCDIS** Debugger ECC Read Disable

| Value | Description |
|---|---|
| 0 | ECC errors for debugger reads are corrected and logged in INTFLAG |
| 1 | ECC errors for debugger reads are not corrected or logged in INTFLAG |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.13. SmartEEPROM Configuration

**Name:** SEECFG
**Offset:** 0x2A
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | APRDIS | WMODE |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – APRDIS** Automatic Page Reallocation Disable
0: enables the Automatic page Reallocation.
1: disables the Automatic page Reallocation.

**Bit 0 – WMODE** Write Mode
Indicates the type of bufferization used.

| Value | Name | Description |
|---|---|---|
| 0x0 | UNBUFFERED | A NVM write command is issued after each write in the pagebuffer |
| 0x1 | BUFFERED | A NVM write command is issued when a write to a new page is requested |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 25.8.14. SmartEEPROM Status

**Name:** SEESTAT
**Offset:** 0x2C
**Reset:** 0x000x0x0x
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | PSZ[2:0] | | |
| Access | | | | | | R | R | R |
| Reset | | | | | | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SBLK[3:0] | | | |
| Access | | | | | R | R | R | R |
| Reset | | | | | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | RLOCK | LOCK | BUSY | LOAD | ASEES |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | x | 0 | 0 | x |

**Bits 18:16 – PSZ[2:0]** SmartEEPROM Page Size
This bit field is automatically loaded from the user page during startup, refer to the Memories.
Indicates the page size.

**Bits 11:8 – SBLK[3:0]** Blocks Number In a Sector
This bit field is automatically loaded from the user page during startup, refer to the Memories.
Indicates the number of blocks allocated to a SEES.

**Bit 4 – RLOCK** RLOCK
SmartEEPROM Write Access To Register Address Space Is Locked

**Bit 3 – LOCK** SmartEEPROM Section Locked
This bit field is automatically loaded from the user page during startup.
Access to the SmartEEPROM data is locked. Writes to AHB2 throws hardfault exceptions.
0: SmartEEPROM access is not locked
1: SmartEEPROM access is locked

**Bit 2 – BUSY** Busy
0: SmartEEPROM is ready.
1: SmartEEPROM is busy processing a read or a write operation.

**Bit 1 – LOAD** Page Buffer Loaded
0: SmartEEPROM has not left unwritten data in the page buffer.
1: SmartEEPROM has left unwritten data in the page buffer.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – ASEES** Active SmartEEPROM Sector

This bit field is automatically loaded during startup from a special fuse in the NVM.
Indicates the active SEES
0: SEES0 is active
1: SEES1 is active


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 26. ICM - Integrity Check Monitor

## 26.1. Overview

The Integrity Check Monitor (ICM) is a DMA controller that performs hash calculation over multiple memory regions using transfer descriptors located in memory (ICM Descriptor Area). The Hash function is based on the Secure Hash Algorithm (SHA). The ICM controller integrates two modes of operation. The first mode is used to hash a list of memory regions and save the digests to memory (ICM Hash Area). The second mode is an active monitoring of the memory. In this mode, the hash function is evaluated and compared to the digest located at a predefined memory address (ICM Hash Area). If a mismatch occurs, an interrupt is raised.

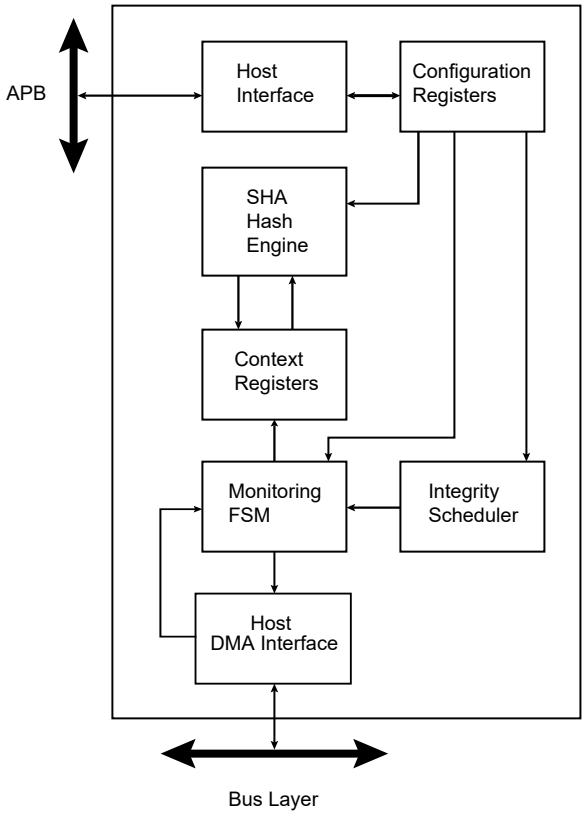
## 26.2. Features

- DMA AHB Host interface
- Supports monitoring of up to four non-contiguous memory regions
- Supports block gathering using a linked list
- Supports Secure Hash Algorithm (SHA1, SHA224, SHA256)
- Compliant with FIPS Publication 180-2
- Configurable processing period:
  - When SHA1 algorithm is processed, the run-time period is either 85 or 209 clock cycles
  - When SHA256 or SHA224 algorithm is processed, the run-time period is either 72 or 194 clock cycles
- Programmable bus burden


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 26.3. Block Diagram

**Figure 26-1.** Integrity Check Monitor Block Diagram




## 26.4. Signal Description

Not applicable.

## 26.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 26.5.1. Power Management

The ICM will run only when the source clocks are running, i.e. when the CPU is in Active mode.

### 26.5.2. Clocks

The ICM bus clocks (CLK_ICM_AHB and CLK_ICM_APB) can be enabled and disabled in the Main Clock module (MCLK) by writing the respective bit in the mask registers (MCLK.AHBMASK.ICM and MCLK.APBCMASK.ICM).

The default states of CLK_ICM_AHB and CLK_ICM_APB are given by the reset values of the respective mask registers.

**Related Links**

Register Summary

### 26.5.3. DMA

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.5.4. Interrupts

The ICM has an interrupt line connected to the Interrupt Controller. Handling the ICM interrupt requires programming the interrupt controller before configuring the ICM.

**Related Links**

Nested Vector Interrupt Controller

### 26.5.5. Events

Not applicable.

### 26.5.6. Debug Operation

Not applicable.

## 26.6. Functional Description

### 26.6.1. Overview

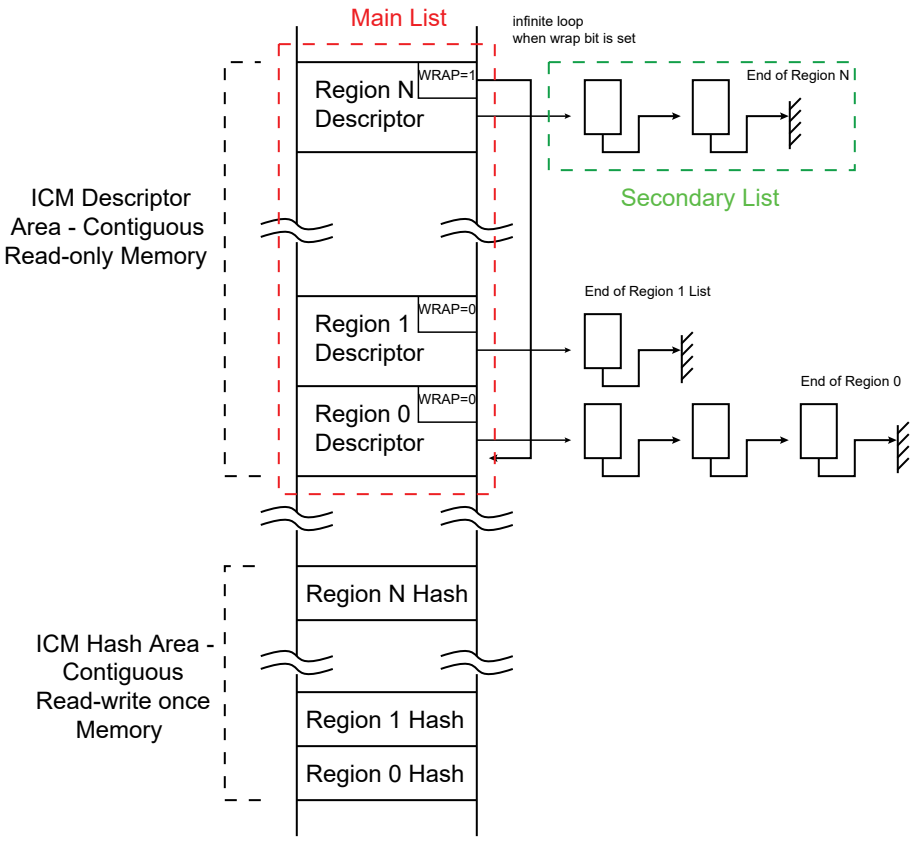
The Integrity Check Monitor (ICM) is a DMA controller that performs SHA-based memory hashing over memory regions. As shown in the Block Diagram, it integrates a DMA interface, a Monitoring Finite State Machine (FSM), an integrity scheduler, a set of context registers, a SHA engine, an interface for configuration and status registers.

The SHA engine requires a message padded according to FIPS180-4 specification when used as a SHA calculation unit only. Otherwise, if the ICM is used as an integrated check for memory content, the padding is not mandatory. The SHA module produces an N-bit message digest each time a block is read and a processing period ends. 'N' is 160 for SHA1, 224 for SHA224, 256 for SHA256.

When the ICM module is enabled, it sequentially retrieves a circular list of region descriptors from the memory (Main List described in the following figure). Up to four regions may be monitored. Each region descriptor is composed of four words indicating the layout of the memory region (refer to the example given in Region Descriptor Structure). It also contains the hashing engine configuration on a per region basis. As soon as the descriptor is loaded from the memory and context registers are updated with the data structure, the hashing operation starts. A programmable number of blocks (see TRSIZE field of the RCTRL structure member) is transferred from the memory to the SHA engine. When the desired number of blocks have been transferred, the digest is either moved to memory (Write Back function) or compared with a digest reference located in the system memory (Compare function). If a digest mismatch occurs, an interrupt is triggered if enabled. The ICM module passes through the region descriptor list until the end of the list marked by an end of list marker (WRAP or EOM bit in ICM_RCFG structure member set to one). To continuously monitor the list of regions, the WRAP bit must be set to one in the last data structure and EOM must be cleared.

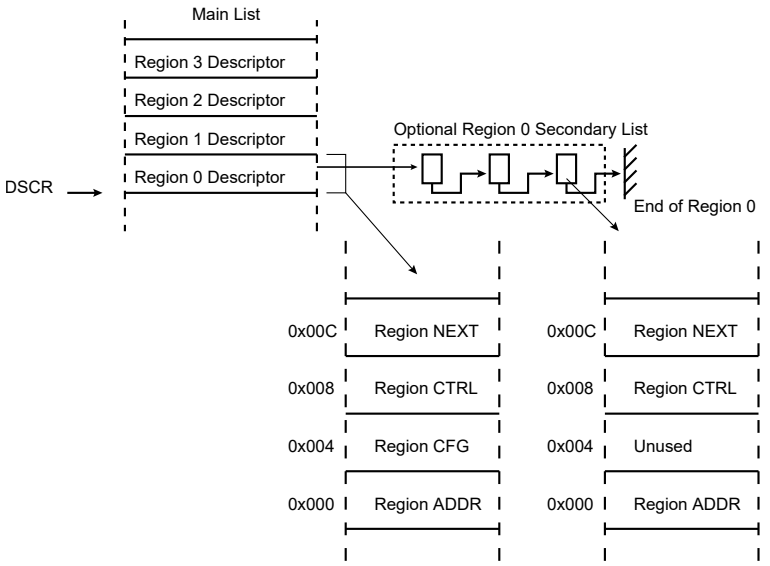

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 26-2.** ICM Region Descriptor and Hash Areas




Each region descriptor supports gathering of data through the use of the Secondary List. Unlike the Main List, the Secondary List cannot modify the configuration attributes of the region. When the end of the Secondary List has been encountered, the ICM returns to the Main List. Memory integrity monitoring can be considered as a background service and the mandatory bandwidth can be very limited. In order to limit the ICM memory bandwidth, use the BBC field of the CFG register to control ICM memory load.

**Figure 26-3.** Region Descriptor





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.6.2. ICM Hash Area

The ICM Hash Area is a contiguous area of system memory that the controller and the processor can access. The physical location is configured in the ICM hash area start address register. This address is a multiple of 128 bytes. If the CDWBN bit of the context register is cleared (i.e., Write Back activated), the ICM controller performs a digest write operation at the following starting location: *(HASH) + (RID<<5), where RID is the current region context identifier. If the CDWBN bit of the context register is set (i.e., Digest Comparison activated), the ICM controller performs a digest read operation at the same address.

#### 26.6.2.1. Message Digest Example

Considering the following 512 bits message (example given in FIPS 180-4):

"61626380000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000018"

The message is written to memory in a Little Endian (LE) system architecture.

| Memory Address | Address Offset / Byte Lane | | | |
|---|---|---|---|---|
| | 0x3 / 31:24 | 0x2 / 23:16 | 0x1 / 15:8 | 0x0 / 7:0 |
| 0x000 | 80 | 63 | 62 | 61 |
| 0x004–0x038 | 00 | 00 | 00 | 00 |
| 0x03C | 18 | 00 | 00 | 00 |

The digest is stored at the memory location pointed at by the ICM_HASH pointer with a Region Offset.

| Memory Address | Address Offset / Byte Lane | | | |
|---|---|---|---|---|
| | 0x3 / 31:24 | 0x2 / 23:16 | 0x1 / 15:8 | 0x0 / 7:0 |
| 0x000 | 36 | 3e | 99 | a9 |
| 0x004 | 6a | 81 | 06 | 47 |
| 0x008 | 71 | 25 | 3e | ba |
| 0x00C | 6c | c2 | 50 | 78 |
| 0x010 | 9d | d8 | d0 | 9c |

| Memory Address | Address Offset / Byte Lane | | | |
|---|---|---|---|---|
| | 0x3 / 31:24 | 0x2 / 23:16 | 0x1 / 15:8 | 0x0 / 7:0 |
| 0x000 | 22 | 7d | 09 | 23 |
| 0x004 | 22 | d8 | 05 | 34 |
| 0x008 | 77 | a4 | 42 | 86 |
| 0x00C | b3 | 55 | a2 | bd |
| 0x010 | e4 | bc | ad | 2a |
| 0x014 | f7 | b3 | a0 | bd |
| 0x018 | a7 | 9d | 6c | e3 |

| Memory Address | Address Offset / Byte Lane | | | |
|---|---|---|---|---|
| | 0x3 / 31:24 | 0x2 / 23:16 | 0x1 / 15:8 | 0x0 / 7:0 |
| 0x000 | bf | 16 | 78 | ba |
| 0x004 | ea | cf | 01 | 8f |
| 0x008 | de | 40 | 41 | 41 |
| 0x00C | 23 | 22 | ae | 5d |
| 0x010 | a3 | 61 | 03 | b0 |
| 0x014 | 9c | 7a | 17 | 96 |

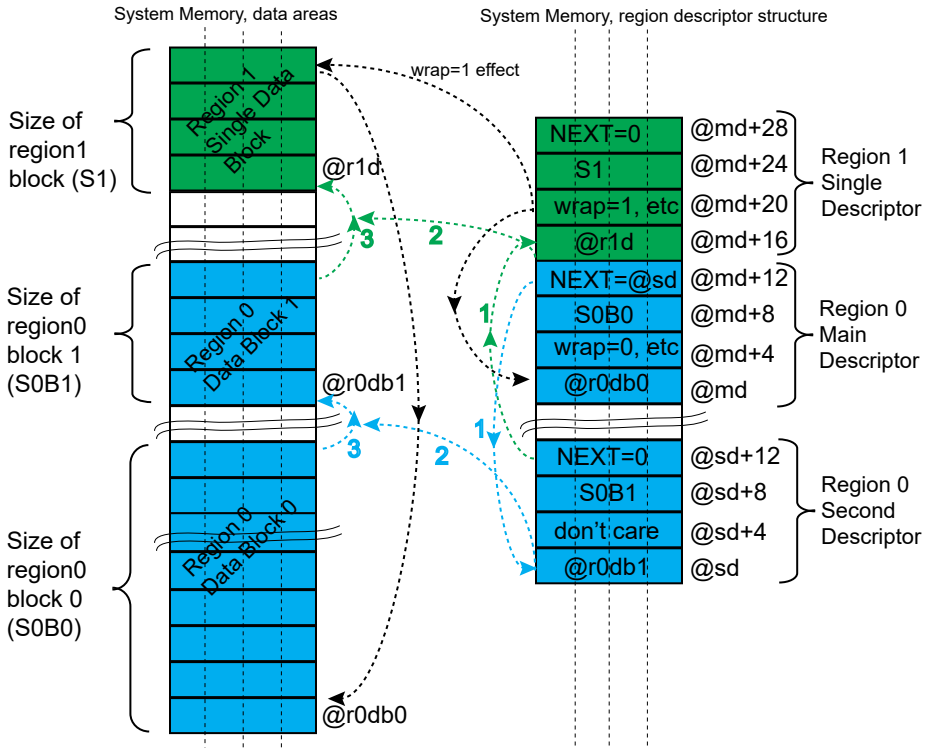

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| **Message Digest Example** (continued) | | | | |
|---|---|---|---|---|
| Memory Address | Address Offset / Byte Lane | | | |
| | 0x3 / 31:24 | 0x2 / 23:16 | 0x1 / 15:8 | 0x0 / 7:0 |
| 0x018 | 61 | ff | 10 | b4 |
| 0x01C | ad | 15 | 00 | f2 |

Considering the following 1024 bits message (example given in FIPS 180-4):

“61626380000000000

**Figure 26-4.** Example - Monitoring of 3 Memory Data Blocks (Defined as 2 Regions)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.6.3.1. Region Descriptor Structure Overview

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | RADDR0 | 7:0 | RADDR[7:0] | | | | | | | |
| | | 15:8 | RADDR[15:8] | | | | | | | |
| | | 23:16 | RADDR[23:16] | | | | | | | |
| | | 31:24 | RADDR[31:24] | | | | | | | |
| 0x04 | RCFG0 | 7:0 | WCIEN | BEIEN | DMIEN | RHIEN | | EOM | WRAP | CDWBN |
| | | 15:8 | | ALGO[2:0] | | | | PROCDLY | SUIEN | ECIEN |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | RCTRL0 | 7:0 | TRSIZE[7:0] | | | | | | | |
| | | 15:8 | TRSIZE[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | RADDR1 | 7:0 | RADDR[7:0] | | | | | | | |
| | | 15:8 | RADDR[15:8] | | | | | | | |
| | | 23:16 | RADDR[23:16] | | | | | | | |
| | | 31:24 | RADDR[31:24] | | | | | | | |
| 0x0C | RNEXT0 | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | RCFG1 | 7:0 | WCIEN | BEIEN | DMIEN | RHIEN | | EOM | WRAP | CDWBN |
| | | 15:8 | | ALGO[2:0] | | | | PROCDLY | SUIEN | ECIEN |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | RCTRL1 | 7:0 | TRSIZE[7:0] | | | | | | | |
| | | 15:8 | TRSIZE[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | RADDR2 | 7:0 | RADDR[7:0] | | | | | | | |
| | | 15:8 | RADDR[15:8] | | | | | | | |
| | | 23:16 | RADDR[23:16] | | | | | | | |
| | | 31:24 | RADDR[31:24] | | | | | | | |
| 0x18 | RNEXT1 | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | RCFG2 | 7:0 | WCIEN | BEIEN | DMIEN | RHIEN | | EOM | WRAP | CDWBN |
| | | 15:8 | | ALGO[2:0] | | | | PROCDLY | SUIEN | ECIEN |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | RCTRL2 | 7:0 | TRSIZE[7:0] | | | | | | | |
| | | 15:8 | TRSIZE[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | RADDR3 | 7:0 | RADDR[7:0] | | | | | | | |
| | | 15:8 | RADDR[15:8] | | | | | | | |
| | | 23:16 | RADDR[23:16] | | | | | | | |
| | | 31:24 | RADDR[31:24] | | | | | | | |
| 0x24 | RNEXT2 | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | RCFG3 | 7:0 | WCIEN | BEIEN | DMIEN | RHIEN | | EOM | WRAP | CDWBN |
| | | 15:8 | | ALGO[2:0] | | | | PROCDLY | SUIEN | ECIEN |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Region Descriptor Structure Overview** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x2C | RCTRL3 | 7:0 | TRSIZE[7:0] | | | | | | | |
| | | 15:8 | TRSIZE[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x30 | RNEXT3 | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 26.6.3.1.1. Region Start Address Structure Member

**Name:** RADDR
**Offset:** 0x00 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RADDR[31:0]** Region Start Address
This field indicates the first byte address of the region


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.6.3.1.2. Region Configuration Structure Member

**Name:** RCFG
**Offset:** 0x04 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | ALGO[2:0] | | | | PROCDLY | SUIEN | ECIEN |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WCIEN | BEIEN | DMIEN | RHIEN | | EOM | WRAP | CDWBN |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | | 0 | 0 | 0 |

**Bits 14:12 – ALGO[2:0]** User SHA Algorithm

| Value | Name | Description |
|---|---|---|
| 0 | SHA1 | SHA1 algorithm processed |
| 1 | SHA256 | SHA256 algorithm processed |
| 4 | SHA224 | SHA224 algorithm processed |
| Other | - | Reserved |

**Bit 10 – PROCDLY** Processing Delay

For a given SHA algorithm, the runtime period has two possible lengths:

**Table 26-2.** SHA Processing Runtime Periods

| Algorithm | SHORTEST [number of cycles] | LONGEST [number of cycles] |
|---|---|---|
| SHA1 | 85 | 209 |
| SHA224 | 72 | 194 |
| SHA256 | 72 | 194 |

| Value | Name | Description |
|---|---|---|
| 0 | SHORTEST | SHA processing runtime is the shortest one |
| 1 | LONGEST | SHA processing runtime is the longest one |

**Bit 9 – SUIEN** Monitoring Status Updated Condition Interrupt Enable

0: The RSU flag is set when the corresponding descriptor is loaded from memory to ICM.
1: The RSU flag remains cleared even if the condition is met.

**Bit 8 – ECIEN** End Bit Condition Interrupt Enable

0: The REC flag is set when the descriptor having the EOM bit set is processed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

1: The REC flag remains cleared even if the setting condition is met.

**Bit 7 – WCIEN** Wrap Condition Interrupt Disable
0: The RWC flag is set when the WRAP
1: The RWC flag remains cleared even if the setting condition is met.

**Bit 6 – BEIEN** Bus Error Interrupt Disable
0: The flag is set when an error is reported on the system bus by the bus MATRIX.
1: The flag remains cleared even if the setting condition is met.

**Bit 5 – DMIEN** Digest Mismatch Interrupt Disable
0: The RBE flag is set when the hash value just calculated from the processed region dffers from expected hash value.
1: The RBE flag remains cleared even if the setting condition is met.

**Bit 4 – RHIEN** Region Hash Completed Interrupt Disable
0: The RHC flag is set when the field NEXT = 0 in a descriptor of the main or second list.
1: The RHC flag remains cleared even if the setting condition is met.

**Bit 2 – EOM** End of Monitoring
0: The current descriptor does not terminate the monitoring.
1: The current descriptor terminates the Main List. WRAP bit value has no effect.

**Bit 1 – WRAP** Wrap Command
0: The next region descriptor address loaded is the current region identifier descriptor address incremented by 0x10.
1: The next region descriptor address loaded is DSCR.

**Bit 0 – CDWBN** Compare Digest or Write Back Digest
0: The digest is written to the Hash area.
1: The digest value is compared to the digest stored in the Hash area.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 26.6.3.1.3. Region Control Structure Member

**Name:** RCTRL
**Offset:** 0x08 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** R/W

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TRSIZE[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TRSIZE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – TRSIZE[15:0]** Transfer Size for the Current Chunk of Data


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 26.6.3.1.4. Region Next Address Structure Member

**Name:** RNEXT
**Offset:** 0x0C + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

## 26.6.4. Using ICM as an SHA Engine

The ICM can be configured to only calculate a SHA1, SHA224, SHA256 digest value.

### 26.6.4.1. Settings for Simple SHA Calculation

The start address of the system memory containing the data to hash must be configured in the transfer descriptor of the DMA embedded in the ICM.

The transfer descriptor is a system memory area integer multiple of 4 x 32-bit word and the start address of the descriptor must be configured in DSCR (the start address must be aligned on 64-bytes; six LSB must be cleared). If the data to hash is already padded according to SHA standards, only a single descriptor is required, and the EOM bit of RCFGn must be written to 1. If the data to hash does not contain a padding area, it is possible to define the padding area in another system memory location, the ICM can be configured to automatically jump from a memory area to another one by writing the descriptor register RNEXT with a value that differs from 0. Writing the RNEXT register with the start address of the padding area forces the ICM to concatenate both areas, thus providing the SHA result from the start address of the hash area configured in HASH.

Whether the system memory is configured as a single or multiple data block area, the bits CDWBN and WRAP must be cleared in the region descriptor structure member RCFGn. The bits WBDIS, EOMDIS, SLBDIS must be cleared in CFG.

Write the bits RHIEN and ECIEN in the Region Configuration Structure Member (RCFGn) to '0':

- The flag RHC[i], i being the region index, is set (if RHIEN is '0') when the hash result is available at address defined in HASH.
- The flag REC[i], i being the region index, is set (if ECIEN is '0') when the hash result is available at the address defined in HASH.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

An interrupt is generated if the bit RHC[i] is written to '1' in the IER (if RHC[i] is set in RCTRL of region i) or if the bit REC[i] is written to 1 in the IER (if REC[i] is set in RCTRL of region i).

#### 26.6.4.2. Processing Period

The SHA engine processing period can be configured by writing to the Region Configuration Structure Member register (RCFGn).

The short processing period allows to allocate bandwidth to the SHA module whereas the long processing period allocates more bandwidth on the system bus to other applications.

In SHA mode, the shortest processing period is 85 clock cycles + 2 clock cycles for start command synchronization. The longest period is 209 clock cycles + 2 clock cycles.

In SHA256 and SHA224 modes, the shortest processing period is 72 clock cycles + 2 clock cycles for start command synchronization. The longest period is 194 clock cycles + 2 clock cycles.

### 26.6.5. ICM Automatic Monitoring Mode

The ASCD bit of the CFG register is used to activate the ICM Automatic Mode. When CFG.ASCD is set, the ICM performs the following actions:

- The ICM controller passes through the Main List once with CDWBN bit in RCFGn at 0 (i.e., Write Back activated) and EOM bit in the RCFGn context register at 0.
- When RCFGn.WRAP=1, the ICM controller enters active monitoring, with CDWBN bit in context register now set, and EOM bit in context register cleared. Writing to the CDWBN and EOM bits in RCFGn has no effect.

### 26.6.6. ICM Configuration Parameters

| Transfer Type | | Main List | RCFG | | | RNEXT | Comments |
|---|---|---|---|---|---|---|---|
| | | | CDWBN | WRAP | EOM | NEXT | |
| Single Region | Contiguous list of blocks<br><br>Digest written to memory<br><br>Monitoring disabled | 1 item | 0 | 0 | 1 | 0 | The Main List contains only one descriptor. The Secondary List is empty for that descriptor. The digest is computed and saved to memory. |
| | Non-contiguous list of blocks<br><br>Digest written to memory<br><br>Monitoring disabled | 1 item | 0 | 0 | 1 | Secondary List address of the current region identifier | The Main List contains only one descriptor. The Secondary List describes the layout of the non-contiguous region. |
| | Contiguous list of blocks<br>Digest comparison enabled<br>Monitoring enabled | 1 item | 1 | 1 | 0 | 0 | When the hash computation is terminated, the digest is compared with the one saved in memory. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**ICM Configuration Parameters** (continued)

| Transfer Type | | Main List | RCFG | | | RNEXT | Comments |
|---|---|---|---|---|---|---|---|
| | | | CDWBN | WRAP | EOM | NEXT | |
| Multiple Regions | Contiguous list of blocks<br>Digest written to memory<br>Monitoring disabled | More than one item | 0 | 0 | 1 for the last, 0 otherwise | 0 | ICM passes through the list once. |
| | Contiguous list of blocks<br><br>Digest comparison is enabled<br><br>Monitoring is enabled | More than one item | 1 | 1 for the last, 0 otherwise | 0 | 0 | ICM performs active monitoring of the regions. If a mismatch occurs, an interrupt is raised. |
| | Non-contiguous list of blocks<br>Digest is written to memory<br>Monitoring is disabled | More than one item | 0 | 0 | 1 | Secondary List address | ICM performs hashing and saves digests to the Hash area. |
| | Non-contiguous list of blocks<br>Digest comparison is enabled<br><br>Monitoring is enabled | More than one item | 1 | 1 | 0 | Secondary List address | ICM performs data gathering on a per region basis. |

### 26.6.7. Security Features

When an undefined register access occurs, the URAD bit in the Interrupt Status Register (ISR) is set if unmasked. Its source is then reported in the Undefined Access Status Register (UASR). Only the first undefined register access is available through the UASR.URAT field.

Several kinds of unspecified register accesses can occur:

- Unspecified structure member set to one detected when the descriptor is loaded
- Configuration register (CFG) modified during active monitoring
- Descriptor register (DSCR) modified during active monitoring
- Hash register (HASH) modified during active monitoring
- Write-only register read access

The URAD bit and the URAT field can only be reset by writing a 1 to the CTRL.SWRST bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 26.7. Register Summary - ICM

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CFG | 7:0 | BBC[3:0] | | | | | SLBDIS | EOMDIS | WBDIS |
| | | 15:8 | UALGO[2:0] | | | UIHASH | | | DUALBUFF | ASCD |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CTRL | 7:0 | REHASH[3:0] | | | | | SWRST | DISABLE | ENABLE |
| | | 15:8 | RMEN[3:0] | | | | RMDIS[3:0] | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | SR | 7:0 | | | | | | | | ENABLE |
| | | 15:8 | RMDIS[3:0] | | | | RAWRMDIS[3:0] | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | IER | 7:0 | RDM[3:0] | | | | RHC[3:0] | | | |
| | | 15:8 | RWC[3:0] | | | | RBE[3:0] | | | |
| | | 23:16 | RSU[3:0] | | | | REC[3:0] | | | |
| | | 31:24 | | | | | | | | URAD |
| 0x14 | IDR | 7:0 | RDM[3:0] | | | | RHC[3:0] | | | |
| | | 15:8 | RWC[3:0] | | | | RBE[3:0] | | | |
| | | 23:16 | RSU[3:0] | | | | REC[3:0] | | | |
| | | 31:24 | | | | | | | | URAD |
| 0x18 | IMR | 7:0 | RDM[3:0] | | | | RHC[3:0] | | | |
| | | 15:8 | RWC[3:0] | | | | RBE[3:0] | | | |
| | | 23:16 | RSU[3:0] | | | | REC[3:0] | | | |
| | | 31:24 | | | | | | | | URAD |
| 0x1C | ISR | 7:0 | RDM[3:0] | | | | RHC[3:0] | | | |
| | | 15:8 | RWC[3:0] | | | | RBE[3:0] | | | |
| | | 23:16 | RSU[3:0] | | | | REC[3:0] | | | |
| | | 31:24 | | | | | | | | URAD |
| 0x20 | UASR | 7:0 | | | | | | URAT[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DSCR | 7:0 | DASA[1:0] | | | | | | | |
| | | 15:8 | DASA[9:2] | | | | | | | |
| | | 23:16 | DASA[17:10] | | | | | | | |
| | | 31:24 | DASA[25:18] | | | | | | | |
| 0x34 | HASH | 7:0 | HASA[0] | | | | | | | |
| | | 15:8 | HASA[8:1] | | | | | | | |
| | | 23:16 | HASA[16:9] | | | | | | | |
| | | 31:24 | HASA[24:17] | | | | | | | |
| 0x38 | UIHVALx0 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |
| 0x3C | UIHVALx1 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |
| 0x40 | UIHVALx2 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary - ICM** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x44 | UIHVALx3 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |
| 0x48 | UIHVALx4 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |
| 0x4C | UIHVALx5 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |
| 0x50 | UIHVALx6 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |
| 0x54 | UIHVALx7 | 7:0 | VAL[7:0] | | | | | | | |
| | | 15:8 | VAL[15:8] | | | | | | | |
| | | 23:16 | VAL[23:16] | | | | | | | |
| | | 31:24 | VAL[31:24] | | | | | | | |

## 26.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

**Related Links**

Region Descriptor Structure


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.8.1. Configuration Register

**Name:** CFG
**Offset:** 0x00
**Reset:** 0x0
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | UALGO[2:0] | | | UIHASH | | | DUALBUFF | ASCD |
| Access | - | - | - | - | | | - | R/W |
| Reset | 0 | 0 | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BBC[3:0] | | | | | SLBDIS | EOMDIS | WBDIS |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 15:13 – UALGO[2:0]** User SHA Algorithm

| Value | Name | Description |
|---|---|---|
| 0 | SHA1 | SHA1 algorithm processed |
| 1 | SHA256 | SHA256 algorithm processed |
| 4 | SHA224 | SHA224 algorithm processed |
| Other | - | Reserved |

**Bit 12 – UIHASH** User Initial Hash Value

| Value | Description |
|---|---|
| 0 | The secure hash standard provides the initial hash value. |
| 1 | The initial hash value is programmable. Field UALGO provides the SHA algorithm. The ALGO field of the RCFGn structure member has no effect. |

**Bit 9 – DUALBUFF** Dual Input Buffer

| Value | Description |
|---|---|
| 0 | Dual Input buffer mode is disabled. |
| 1 | Dual Input buffer mode is enabled (Better performances, higher bandwidth required on system bus). |

**Bit 8 – ASCD** Automatic Switch To Compare Digest

| Value | Description |
|---|---|
| 0 | Automatic mode is disabled. |
| 1 | When this mode is enabled, the ICM controller automatically switches to active monitoring after the first Main List pass. Both CDWBN and WBDIS bits have no effect. A '1' must be written to the End of Monitoring bit in the Region Configuration register (RCFG.EOM) to terminate the monitoring. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 7:4 – BBC[3:0]** Bus Burden Control

This field is used to control the burden of the ICM system bus. The number of system clock cycles between the end of the current processing and the next block transfer is set to $2^{BBC}$. Up to 32768 cycles can be inserted.

**Bit 2 – SLBDIS** Secondary List Branching Disable

| Value | Description |
|---|---|
| 0 | Branching to the Secondary List is permitted. |
| 1 | Branching to the Secondary List is forbidden. The NEXT field of the RNEXT structure member has no effect and is always considered as zero. |

**Bit 1 – EOMDIS** End of Monitoring Disable

| Value | Description |
|---|---|
| 0 | End of Monitoring is permitted. |
| 1 | End of Monitoring is forbidden. The EOM bit of the RCFG structure member has no effect. |

**Bit 0 – WBDIS** Write Back Disable

1:

When the Automatic Switch to Compare Digest bit of this register (CFG.ASCD) is written to '1', this bit value has no effect.

| Value | Description |
|---|---|
| 0 | Write Back Operations are permitted. |
| 1 | Write Back Operations are forbidden: Context register CDWBN bit is internally set to '1' and cannot be modified by a linked list element. The CDWBN bit of the RCFG structure member has no effect. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.8.2. Control Register

**Name:** CTRL
**Offset:** 0x04
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RMEN[3:0] | | | | RMDIS[3:0] | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | REHASH[3:0] | | | | | SWRST | DISABLE | ENABLE |
| Access | W | W | W | W | | W | W | W |
| Reset | | | | | | | 0 | 0 |

**Bits 15:12 – RMEN[3:0]** Region Monitoring Enable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When bit RMEN[i] is written to '1', the monitoring of region with identifier i is activated. |

**Bits 11:8 – RMDIS[3:0]** Region Monitoring Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When REHASH[i] is written to '1', Region i digest is re-computed. This bit is only available when region monitoring is disabled. |

**Bits 7:4 – REHASH[3:0]** Recompute Internal Hash

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When REHASH[i] is written to '1', Region i digest is re-computed. This bit is only available when region monitoring is disabled. |

**Bit 2 – SWRST** Software Reset

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Resets the ICM controller. |

**Bit 1 – DISABLE** ECM Disable

| Value | Description |
|---|---|
| 0 | No effect. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The ICM controller is disabled. If a region is activated, the region is terminated. |

**Bit 0 – ENABLE** ICM Enable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | The ICM controller is activated. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 26.8.3. Status Register

**Name:** SR
**Offset:** 0x08
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RMDIS[3:0] | | | | RAWRMDIS[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | ENABLE |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

**Bits 15:12 – RMDIS[3:0]** Region Monitoring Disabled Status

| Value | Description |
|---|---|
| 0 | Region i is being monitored (occurs after integrity check value has been calculated and written to Hash area). |
| 1 | Region i is not being monitored. |

**Bits 11:8 – RAWRMDIS[3:0]** Region Monitoring Disabled Raw Status

| Value | Description |
|---|---|
| 0 | Region i monitoring has been activated by writing a 1 in RMEN[i] of CTRL |
| 1 | Region i monitoring has been deactivated by writing a 1 in RMDIS[i] of CTRL |

**Bit 0 – ENABLE** ICM Controller Enable Register

| Value | Description |
|---|---|
| 0 | ICM controller is disabled. |
| 1 | ICM controller is activated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.8.4. Interrupt Enable Register

**Name:** IER
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** Write-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | URAD |
| Access | | | | | | | | W |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RSU[3:0] | | | | REC[3:0] | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RWC[3:0] | | | | RBE[3:0] | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RDM[3:0] | | | | RHC[3:0] | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 24 – URAD** Undefined Register Access Detection Interrupt Enable
0: No effect
1: The Undefined Register Access interrupt is enabled.

**Bits 23:20 – RSU[3:0]** Region Status Updated Interrupt Enable
0: No effect
1: When RSU[i] is written to '1', the region i Status Updated interrupt is enabled.

**Bits 19:16 – REC[3:0]** Region End bit Condition Detected Interrupt Enable
0: No effect
1: When REC[i] is written to '1', the region i End bit Condition interrupt is enabled.

**Bits 15:12 – RWC[3:0]** Region Wrap Condition detected Interrupt Enable
0: No effect
1: When RWC[i] is written to '1', the Region i Wrap Condition interrupt is enabled.

**Bits 11:8 – RBE[3:0]** Region Bus Error Interrupt Enable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RBE[i] is written to '1', the Region i Bus Error interrupt is enabled. |

**Bits 7:4 – RDM[3:0]** Region Digest Mismatch Interrupt Enable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RDM[i] is written to '1', the Region i Digest Mismatch interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 3:0 – RHC[3:0]** Region Hash Completed Interrupt Enable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RHC[i] is written to '1', the Region i Hash Completed interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 26.8.5. Interrupt Disable Register

**Name:** IDR
**Offset:** 0x14
**Property:** Write-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | URAD |
| Access | | | | | | | | W |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RSU[3:0] | | | | REC[3:0] | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RWC[3:0] | | | | RBE[3:0] | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RDM[3:0] | | | | RHC[3:0] | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | | | | | | | | |

**Bit 24 – URAD** Undefined Register Access Detection Interrupt Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Undefined Register Access Detection interrupt is disabled. |

**Bits 23:20 – RSU[3:0]** Region Status Updated Interrupt Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RSU[i] is written to '1', the region i Status Updated interrupt is disabled. |

**Bits 19:16 – REC[3:0]** Region End bit Condition detected Interrupt Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When REC[i] is written to '1', the region i End bit Condition interrupt is disabled. |

**Bits 15:12 – RWC[3:0]** Region Wrap Condition Detected Interrupt Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RWC[i] is written to '1', the Region i Wrap Condition interrupt is disabled. |

**Bits 11:8 – RBE[3:0]** Region Bus Error Interrupt Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RBE[i] is written to '1', the Region i Bus Error interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 7:4 – RDM[3:0]** Region Digest Mismatch Interrupt Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RDM[i] is written to '1', the Region i Digest Mismatch interrupt is disabled. |

**Bits 3:0 – RHC[3:0]** Region Hash Completed Interrupt Disable

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When RHC[i] is written to '1', the Region i Hash Completed interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 26.8.6. Interrupt Mask Register

**Name:** IMR
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | URAD |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RSU[3:0] | | | | REC[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RWC[3:0] | | | | RBE[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RDM[3:0] | | | | RHC[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 24 – URAD** Undefined Register Access Detection Interrupt Mask

| Value | Description |
|---|---|
| 0 | The interrupt is disabled. |
| 1 | The interrupt is enabled. |

**Bits 23:20 – RSU[3:0]** Region Status Updated Interrupt Mask

| Value | Description |
|---|---|
| 0 | When RSU[i] is reading '0', the interrupt is disabled for region i. |
| 1 | When RSU[i] is reading '1', the interrupt is enabled for region i. |

**Bits 19:16 – REC[3:0]** Region End bit Condition Detected Interrupt Mask

| Value | Description |
|---|---|
| 0 | When REC[i] is reading '0', the interrupt is disabled for region i. |
| 1 | When REC[i] is reading '1', the interrupt is enabled for region i. |

**Bits 15:12 – RWC[3:0]** Region Wrap Condition Detected Interrupt Mask

| Value | Description |
|---|---|
| 0 | When RWC[i] is reading '0', the interrupt is disabled for region i. |
| 1 | When RWC[i] is reading '1', the interrupt is enabled for region i. |

**Bits 11:8 – RBE[3:0]** Region Bus Error Interrupt Mask

| Value | Description |
|---|---|
| 0 | When RBE[i] is reading '0', the interrupt is disabled for region i. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | When RBE[i] is reading '1', the interrupt is enabled for region i. |

**Bits 7:4 – RDM[3:0]** Region Digest Mismatch Interrupt Mask

| Value | Description |
|---|---|
| 0 | When RDM[i] is reading '0', the interrupt is disabled for region i. |
| 1 | When RDM[i] is reading '1', the interrupt is enabled for region i. |

**Bits 3:0 – RHC[3:0]** Region Hash Completed Interrupt Mask

| Value | Description |
|---|---|
| 0 | When RHC[i] is reading '0', the interrupt is disabled for region i. |
| 1 | When RHC[i] is reading '1', the interrupt is enabled for region i. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 26.8.7. Interrupt Status Register

**Name:** ISR
**Offset:** 0x1C
**Reset:** 0x0
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | URAD |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RSU[3:0] | | | | REC[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RWC[3:0] | | | | RBE[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RDM[3:0] | | | | RHC[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 24 – URAD** Undefined Register Access Detection Status
The URAD bit is only reset by the SWRST bit in the CTRL register.
The Undefined Register Access Trace bit field in the Undefined Access Status Register (UASR.URAT) indicates the unspecified access type.

| Value | Description |
|---|---|
| 0 | No undefined register access has been detected since the last SWRST. |
| 1 | At least one undefined register access has been detected since the last SWRST. |

**Bits 23:20 – RSU[3:0]** Region Status Updated Detected
RSU[i] is set when a region status updated condition is detected.

**Bits 19:16 – REC[3:0]** Region End bit Condition Detected
REC[i] is set when an end bit condition is detected.

**Bits 15:12 – RWC[3:0]** Region Wrap Condition Detected
RWC[i] is set when a wrap condition is detected.

**Bits 11:8 – RBE[3:0]** Region Bus Error
RBE[i] is set when a bus error is detected while hashing memory region i.

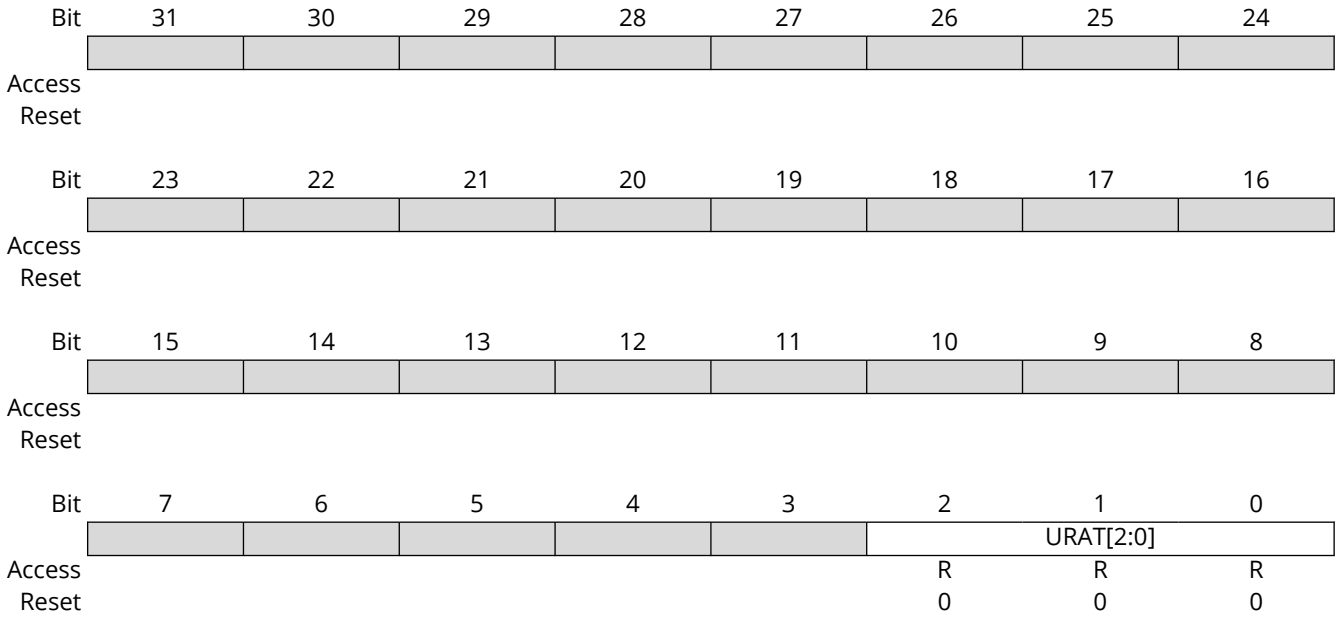
**Bits 7:4 – RDM[3:0]** Region Digest Mismatch
RDM[i] is set when there is a digest comparison mismatch between the hash value of region i and the reference value located in the Hash Area.

**Bits 3:0 – RHC[3:0]** Region Hash Completed
RHC[i] is set when the ICM has completed the region with identifier i.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.8.8. Undefined Access Status Register

**Name:** UASR
**Offset:** 0x20
**Reset:** 0x0
**Property:** Read-Only




**Bits 2:0 – URAT[2:0]** Undefined Register Access Trace
Only the first Undefined Register Access Trace is available through the URAT field.
The URAT field is only reset by the Software Reset bit in the Control register (CTRL.SWRST).

| Value | Name | Description |
|---|---|---|
| 0 | UNSPEC_STRUCT_MEMBER | Unspecified structure member set to '1' detected when the descriptor is loaded. |
| 1 | ICM_CFG_MODIFIED | CFG modified during active monitoring. |
| 2 | ICM_DSCR_MODIFIED | DSCR modified during active monitoring. |
| 3 | ICM_HASH_MODIFIED | HASH modified during active monitoring |
| 4 | READ_ACCESS | Write-only register read access<br>Only the first Undefined Register Access Trace is available through the URAT field.<br>The URAT field is only reset by the SWRST bit in the CTRL register. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.8.9. Descriptor Area Start Address Register

**Name:** DSCR
**Offset:** 0x30
**Reset:** 0x0
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DASA[25:18] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DASA[17:10] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DASA[9:2] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DASA[1:0] | | | | | | | |
| Access | R/W | R/W | | | | | | |
| Reset | 0 | 0 | | | | | | |

**Bits 31:6 – DASA[25:0]** Descriptor Area Start Address
The start address is a multiple of the total size of the data structure (64 bytes).


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 26.8.10. Hash Area Start Address Register

**Name:** HASH
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | HASA[24:17] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | HASA[16:9] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | HASA[8:1] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HASA[0] | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

**Bits 31:7 – HASA[24:0]** Hash Area Start Address
This field points at the Hash memory location. The address must be a multiple of 128 bytes.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 26.8.11. User Initial Hash Value Register

**Name:** UIHVALx
**Offset:** 0x38 + x*0x04 [x=0..7]
**Reset:** 0
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | VAL[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | VAL[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | VAL[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VAL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – VAL[31:0]** Initial Hash Value

When UIHASH bit of CFG register is set, the Initial Hash Value is user-programmable.
To meet the desired standard, use the following example values.
For UIHVAL0 field:

| Example | Comment |
|---|---|
| 0x67452301 | SHA1 algorithm |
| 0xC1059ED8 | SHA224 algorithm |
| 0x6A09E667 | SHA256 algorithm |

For UIHVAL1 field:

| Example | Comment |
|---|---|
| 0xEFCDAB89 | SHA1 algorithm |
| 0x367CD507 | SHA224 algorithm |
| 0xBB67AE85 | SHA256 algorithm |

For UIHVAL2 field:

| Example | Comment |
|---|---|
| 0x98BADCFE | SHA1 algorithm |
| 0x3070DD17 | SHA224 algorithm |
| 0x3C6EF372 | SHA256 algorithm |

For UIHVAL3 field:

| Example | Comment |
|---|---|
| 0x10325476 | SHA1 algorithm |
| 0xF70E5939 | SHA224 algorithm |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| **VAL** (continued) | |
|---|---|
| **Example** | **Comment** |
| 0xA54FF53A | SHA256 algorithm |

For UIHVAL4 field:

| Example | Comment |
|---|---|
| 0xC3D2E1F0 | SHA1 algorithm |
| 0xFFC00B31 | SHA224 algorithm |
| 0x510E527F | SHA256 algorithm |

For UIHVAL5 field:

| Example | Comment |
|---|---|
| 0x68581511 | SHA224 algorithm |
| 0x9B05688C | SHA256 algorithm |

For UIHVAL6 field:

| Example | Comment |
|---|---|
| 0x64F98FA7 | SHA224 algorithm |
| 0x1F83D9AB | SHA256 algorithm |

For UIHVAL7 field:

| Example | Comment |
|---|---|
| 0xBEFA4FA4 | SHA224 algorithm |
| 0x5BE0CD19 | SHA256 algorithm |

Example of Initial Value for SHA-1 Algorithm

| Register Address | Address Offset / Byte Lane | | | |
|---|---|---|---|---|
| | 0x3 / 31:24 | 0x2 / 23:16 | 0x1 / 15:8 | 0x0 / 7:0 |
| 0x000 UIHVAL0 | 01 | 23 | 45 | 67 |
| 0x004 UIHVAL1 | 89 | ab | cd | ef |
| 0x008 UIHVAL2 | fe | dc | ba | 98 |
| 0x00C UIHVAL3 | 76 | 54 | 32 | 10 |
| 0x010 UIHVAL4 | f0 | e1 | d2 | c3 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 27. PAC - Peripheral Access Controller

## 27.1. Overview

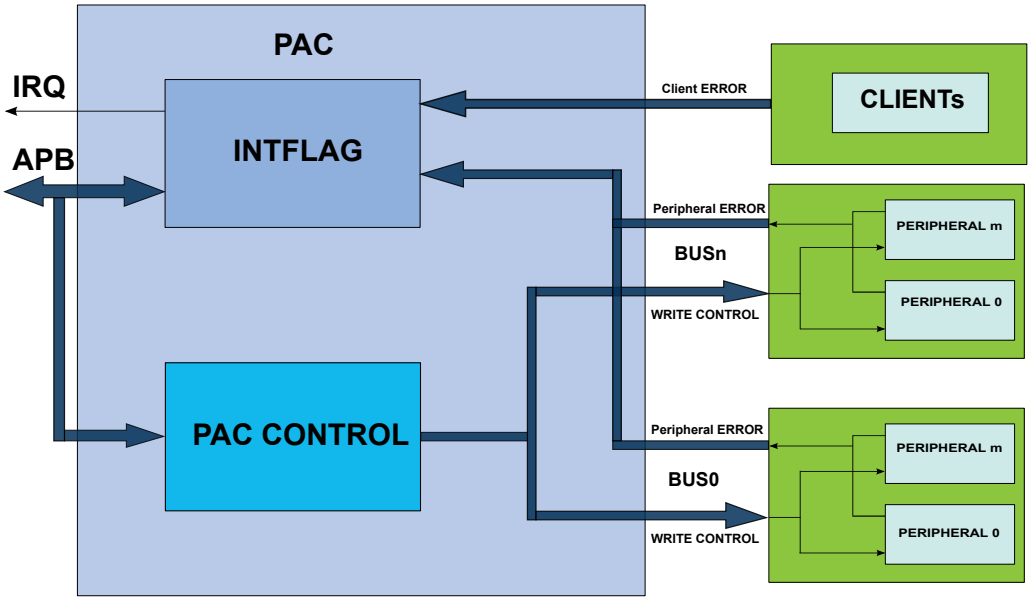
The Peripheral Access Controller provides an interface for the locking and unlocking of peripheral registers within the device. It reports all violations that could happen when accessing a peripheral: write protected access, illegal access, enable protected access, access when clock synchronization or software reset is on-going. These errors are reported in a unique interrupt flag for a peripheral. The PAC module also reports errors occurring at the client bus level, when an access to a non-existing address is detected.

## 27.2. Features

- Manages write protection access and reports access errors for the peripheral modules or bridges.

## 27.3. Block Diagram

**Figure 27-1.** PAC Block Diagram




## 27.4. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 27.4.1. IO Lines

Not applicable.

### 27.4.2. Power Management

The PAC can continue to operate in any Sleep mode where the selected source clock is running. The PAC interrupts can be used to wake up the device from Sleep modes. The events can trigger other operations in the system without exiting sleep modes.

**Related Links**

PM – Power Manager


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.4.3. Clocks

The PAC bus clock (CLK_PAC_APB) can be enabled and disabled in the Main Clock module. The default state of CLK_PAC_APB can be found in the related links.

**Related Links**
MCLK – Main Clock
Peripheral Clock Masking

### 27.4.4. DMA

Not applicable.

### 27.4.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the PAC interrupt requires the Interrupt Controller to be configured first.

**Table 27-1.** Interrupt Lines

| Instances | NVIC Line |
|---|---|
| PAC | ERR |

**Related Links**
Nested Vector Interrupt Controller

### 27.4.6. Events

The events are connected to the Event System, which may need configuration.

**Related Links**
Event System (EVSYS)

### 27.4.7. Debug Operation

When the CPU is halted in Debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

### 27.4.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following PAC registers:

- Write Control (WRCTRL) register
- AHB Client Bus Interrupt Flag Status and Clear (INTFLAGAHB) register
- Peripheral Interrupt Flag Status and Clear n (INTFLAG A/B/C...) registers

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

## 27.5. Functional Description

### 27.5.1. Principle of Operation

The Peripheral Access Control module allows the user to set a write protection on peripheral modules and generate an interrupt in case of a peripheral access violation. The peripheral's protection can be set, cleared or locked at the user discretion. A set of Interrupt Flag and Status registers informs the user on the status of the violation in the peripherals. In addition, client bus errors can be also reported in the cases where reserved area is accessed by the application.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 27.5.2. Basic Operation

### 27.5.2.1. Initialization, Enabling and Resetting

The PAC is always enabled after reset.

Only a hardware reset will reset the PAC module.

### 27.5.2.2. Operations

The PAC module allows the user to set, clear or lock the write protection status of all peripherals on all Peripheral Bridges.

If a peripheral register violation occurs, the Peripheral Interrupt Flag n registers (INTFLAGn) are updated to inform the user on the status of the violation in the peripherals connected to the Peripheral Bridge n (n = A,B,C ...). The corresponding Peripheral Write Control Status n register (STATUSn) gives the state of the write protection for all peripherals connected to the corresponding Peripheral Bridge n. Refer to Peripheral Access Errors for details.

The PAC module also report the errors occurring at client bus level when an access to reserved area is detected. AHB Client Bus Interrupt Flag register (INTFLAGAHB) informs the user on the status of the violation in the corresponding client. Refer to the AHB Client Bus Errors for details.

### 27.5.2.3. Peripheral Access Errors

The following events will generate a Peripheral Access Error:

- Protected write: To avoid unexpected writes to a peripheral's registers, each peripheral can be write protected. Only the registers denoted as “PAC Write-Protection” in the module’s datasheet can be protected. If a peripheral is not write protected, write data accesses are performed normally. If a peripheral is write protected and if a write access is attempted, data will not be written and peripheral returns an access error. The corresponding interrupt flag bit in the INTFLAGn register will be set.
- Illegal access: Access to an unimplemented register within the module.
- Synchronized write error: For write-synchronized registers an error will be reported if the register is written while a synchronization is ongoing.

When any of the INTFLAGn registers bit are set, an interrupt will be requested if the PAC interrupt enable bit is set.

**Related Links**

Register Synchronization

### 27.5.2.4. Write Access Protection Management

Peripheral access control can be enabled or disabled by writing to the WRCTRL register.

The data written to the WRCTRL register is composed of two fields; WRCTRL.PERID and WRCTRL.KEY. The WRCTRL.PERID is an unique identifier corresponding to a peripheral. The WRCTRL.KEY is a key value that defines the operation to be done on the control access bit. These operations can be “clear protection”, “set protection” and “set and lock protection bit”.

The “clear protection” operation will remove the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are allowed for the registers in this peripheral.

The “set protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID. Write accesses are not allowed for the registers with write protection property in this peripheral.

The “set and lock protection” operation will set the write access protection for the peripheral selected by WRCTRL.PERID and locks the access rights of the selected peripheral registers. The write access protection will only be cleared by a hardware reset.

The peripheral access control status can be read from the corresponding STATUSn register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 27.5.2.5. Write Access Protection Management Errors

Only word-wise writes to the WRCTRL register will effectively change the access protection. Other type of accesses will have no effect and will cause a PAC write access error. This error is reported in the INTFLAGx.PAC bit corresponding to the module.

PAC also offers an additional safety feature for correct program execution with an interrupt generated on double write clear protection or double write set protection. If a peripheral is write protected and a subsequent set protection operation is detected then the PAC returns an error, and similarly for a double clear protection operation.

In addition, an error is generated when writing a “set and lock” protection to a write-protected peripheral or when a write access is done to a locked set protection. This can be used to ensure that the application follows the intended program flow by always following a write protect with an unprotect and conversely. However in applications where a write protected peripheral is used in several contexts, e.g. interrupt, care should be taken so that either the interrupt can not happen while the main application or other interrupt levels manipulates the write protection status or when the interrupt handler needs to unprotect the peripheral based on the current protection status by reading the STATUS register.

The errors generated while accessing the PAC module registers (eg. key error, double protect error...) will set the INTFLAGx.PAC flag.

#### 27.5.2.6. AHB Client Bus Errors

The PAC module reports errors occurring at the AHB Client bus level. These errors are generated when an access is performed at an address where no client (bridge or peripheral) is mapped . These errors are reported in the corresponding bits of the INTFLAGAHB register.

#### 27.5.2.7. Generating Events

The PAC module can also generate an event when any of the Interrupt Flag registers bit are set. To enable the PAC event generation, the control bit EVCTRL.ERREO must be set a '1'.

### 27.5.3. DMA Operation

Not applicable.

### 27.5.4. Interrupts

The PAC has the following interrupt source:

- Error (ERR): Indicates that a peripheral access violation occurred in one of the peripherals controlled by the PAC module, or a bridge error occurred in one of the bridges reported by the PAC
  - This interrupt is a synchronous wake-up source

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAGAHB and INTFLAGn) registers is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the PAC is reset. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAGAHB and INTFLAGn registers to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

**Related Links**

Nested Vector Interrupt Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.5.5. Events

The PAC can generate the following output event:

- Error (ERR): Generated when one of the interrupt flag registers bits is set

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.ERREO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

### 27.5.6. Sleep Mode Operation

In Sleep mode, the PAC is kept enabled if an available bus host (CPU, DMA) is running. The PAC will continue to catch access errors from the module and generate interrupts or events.

### 27.5.7. Synchronization

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 27.6. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | WRCTRL | 7:0 | PERID[7:0] | | | | | | | |
| | | 15:8 | PERID[15:8] | | | | | | | |
| | | 23:16 | KEY[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | EVCTRL | 7:0 | | | | | | | | ERREO |
| 0x05 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | | | | | | | | ERR |
| 0x09 | INTENSET | 7:0 | | | | | | | | ERR |
| 0x0A ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | INTFLAGAHB | 7:0 | HPB0 | RAMDMACICM | RAMDMAWR | RAMPPPDSU | RAMCM4S | NVMCTRL2 | NVMCTRL1 | NVMCTRL0 |
| | | 15:8 | | QSPI | SDHC1 | SDHC0 | PUKCC | HPB3 | HPB2 | HPB1 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | INTFLAGA | 7:0 | GCLK | SUPC | OSC32KCTRL | OSCCTRL | RSTC | MCLK | PM | PAC |
| | | 15:8 | TC1 | TC0 | SERCOM1 | SERCOM0 | FREQM | EIC | RTC | WDT |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | INTFLAGB | 7:0 | EVSYS | | DMAC | PORT | CMCC | NVMCTRL | DSU | USB |
| | | 15:8 | | TC3 | TC2 | TCC1 | TCC0 | SERCOM3 | SERCOM2 | |
| | | 23:16 | | | | | | | | RAMECC |
| | | 31:24 | | | | | | | | |
| 0x1C | INTFLAGC | 7:0 | PDEC | TC5 | TC4 | TCC3 | TCC2 | GMAC | CAN1 | CAN0 |
| | | 15:8 | | CCL | QSPI | PUKCC | ICM | TRNG | AES | AC |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | INTFLAGD | 7:0 | ADC0 | TC7 | TC6 | TCC4 | SERCOM7 | SERCOM6 | SERCOM5 | SERCOM4 |
| | | 15:8 | | | | | PCC | I2S | DAC | ADC1 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 ... 0x33 | Reserved | | | | | | | | | |
| 0x34 | STATUSA | 7:0 | GCLK | SUPC | OSC32KCTRL | OSCCTRL | RSTC | MCLK | PM | PAC |
| | | 15:8 | TC1 | TC0 | SERCOM1 | SERCOM0 | FREQM | EIC | RTC | WDT |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x38 | STATUSB | 7:0 | EVSYS | | DMAC | PORT | CMCC | NVMCTRL | DSU | USB |
| | | 15:8 | | TC3 | TC2 | TCC1 | TCC0 | SERCOM3 | SERCOM2 | |
| | | 23:16 | | | | | | | | RAMECC |
| | | 31:24 | | | | | | | | |
| 0x3C | STATUSC | 7:0 | PDEC | TC5 | TC4 | TCC3 | TCC2 | GMAC | CAN1 | CAN0 |
| | | 15:8 | | CCL | QSPI | PUKCC | ICM | TRNG | AES | AC |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x40 | STATUSD | 7:0 | ADC0 | TC7 | TC6 | TCC4 | SERCOM7 | SERCOM6 | SERCOM5 | SERCOM4 |
| | | 15:8 | | | | | PCC | I2S | DAC | ADC1 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 27.7. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the related links.

**Related Links**

Register Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 27.7.1. Write Control

**Name:** WRCTRL
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | KEY[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PERID[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PERID[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – KEY[7:0]** Peripheral Access Control Key
These bits define the peripheral access control key:

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | No action |
| 0x1 | CLEAR | Clear the peripheral write control |
| 0x2 | SET | Set the peripheral write control |
| 0x3 | LOCK | Set and lock the peripheral write control until the next hardware reset |

**Bits 15:0 – PERID[15:0]** Peripheral Identifier
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY. The Peripheral Identifier is calculated following formula:
$PERID = 32^{*}\,\mathrm{BridgeNumber} + \mathrm{N}$

Where BridgeNumber represents the Peripheral Bridge Number (0 for Peripheral Bridge A, 1 for Peripheral Bridge B, etc). N represents the peripheral index from the respective Bridge Number:

**Table 27-2.** PERID Values

| Periph. Bridge Name | BridgeNumber | PERID Values |
|---|---|---|
| A | 0 | 0+N |
| B | 1 | 32+N |
| C | 2 | 64+N |
| D | 3 | 96+N |

**Note:** GMAC, ICM, SDHC, CAN and PCC peripherals do not support that feature.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.7.2. Event Control

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | ERREO |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – ERREO** Peripheral Access Error Event Output
This bit indicates if the Peripheral Access Error Event Output is enabled or disabled. When enabled, an event will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:

| Value | Description |
|---|---|
| 0 | Peripheral Access Error Event Output is disabled. |
| 1 | Peripheral Access Error Event Output is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.7.3. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | ERR |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – ERR** Peripheral Access Error Interrupt Disable
This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Peripheral Access Error interrupt Enable bit and disables the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | Peripheral Access Error interrupt is disabled. |
| 1 | Peripheral Access Error interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.7.4. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | ERR |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – ERR** Peripheral Access Error Interrupt Enable
This bit indicates that the Peripheral Access Error Interrupt is enabled and an interrupt request will be generated when one of the interrupt flag registers bits (INTFLAGAHB, INTFLAGn) is set:
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Peripheral Access Error interrupt Enable bit and enables the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | Peripheral Access Error interrupt is disabled. |
| 1 | Peripheral Access Error interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.7.5. Bridge Interrupt Flag Status

**Name:** INTFLAGAHB
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

These flags are cleared by writing a '1' to the corresponding bit.

These flags are set when an access error is detected by the corresponding AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | QSPI | SDHC1 | SDHC0 | PUKCC | HPB3 | HPB2 | HPB1 |
| Access | | RW | RW | RW | RW | RW | RW | RW |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HPB0 | RAMDMACICM | RAMDMAWR | RAMPPPDSU | RAMCM4S | NVMCTRL2 | NVMCTRL1 | NVMCTRL0 |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 14 – QSPI** Interrupt Flag for QSPI
This flag is set when an access error is detected by the QSPI AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the QSPI interrupt flag.

**Bit 13 – SDHC1** Interrupt Flag for SDHC1
This flag is set when an access error is detected by the SDHC1 AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the SDHC1 interrupt flag.

**Bit 12 – SDHC0** Interrupt Flag for SDHC0
This flag is set when an access error is detected by the SDHC0 AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the SDHC0 interrupt flag.

**Bit 11 – PUKCC** Interrupt Flag for PUKCC
This flag is set when an access error is detected by the PUKCC AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit will clear the PUKCC interrupt flag.

**Bit 10 – HPB3** Interrupt Flag for AHB-APB Bridge D
This flag is set when an access error is detected by the AHB-APB Bridge D Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the HPB3 interrupt flag.

**Bit 9 – HPB2** Interrupt Flag for AHB-APB Bridge C
This flag is set when an access error is detected by the AHB-APB Bridge C Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the HPB2 interrupt flag.

**Bit 8 – HPB1** Interrupt Flag for AHB-APB Bridge B
This flag is set when an access error is detected by the AHB-APB Bridge B Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the HPB1 interrupt flag.

**Bit 7 – HPB0** Interrupt Flag for AHB-APB Bridge A
This flag is set when an access error is detected by the AHB-APB Bridge A Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the HPB0 interrupt flag.

**Bit 6 – RAMDMACICM** Interrupt Flag for RAMDMACICM
This flag is set when an access error is detected by the RAMDMACICM AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the RAMDMACICM interrupt flag.

**Bit 5 – RAMDMAWR** Interrupt Flag for RAMDMAWR
This flag is set when an access error is detected by the RAMDMAWR AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the RAMDMAWR interrupt flag.

**Bit 4 – RAMPPPDSU** Interrupt Flag for RAMPPPDSU:
This flag is set when an access error is detected by the RAMPPPDSU AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the RAMPPPDSU interrupt flag.

**Bit 3 – RAMCM4S** Interrupt Flag for RAMCM4S
This flag is set when an access error is detected by the RAMCM4S AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the RAMCM4S interrupt flag.

**Bit 2 – NVMCTRL2** Interrupt Flag for NVMCTRL2
This flag is set when an access error is detected by the NVMCTRL2 AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the NVMCTRL2 interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 1 – NVMCTRL1** Interrupt Flag for NVMCTRL1
This flag is set when an access error is detected by the NVMCTRL1 AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the NVMCTRL1 interrupt flag.

**Bit 0 – NVMCTRL0** Interrupt Flag for NVMCTRL0
This flag is set when an access error is detected by the NVMCTRL0 AHB Client, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' has no effect.
Writing a '1' to this bit will clear the NVMCTRL0 interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 27.7.6. Peripheral Interrupt Flag Status - Bridge A

**Name:** INTFLAGA
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** –

These flags are set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGx bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits will clear the corresponding INTFLAGx interrupt flag.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TC1 | TC0 | SERCOM1 | SERCOM0 | FREQM | EIC | RTC | WDT |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GCLK | SUPC | OSC32KCTRL | OSCCTRL | RSTC | MCLK | PM | PAC |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – TC1** Interrupt Flag for TC1
This bit is set when a Peripheral Access Error occurs while accessing the TC1, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 14 – TC0** Interrupt Flag for TC0
This bit is set when a Peripheral Access Error occurs while accessing the TC0, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 13 – SERCOM1** Interrupt Flag for SERCOM1
This bit is set when a Peripheral Access Error occurs while accessing the SERCOM1, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 12 – SERCOM0** Interrupt Flag for SERCOM0
This bit is set when a Peripheral Access Error occurs while accessing the SERCOM0, and will generate an interrupt request if SET.ERR is '1'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 11 – FREQM** Interrupt Flag for FREQM
This bit is set when a Peripheral Access Error occurs while accessing the FREQM, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 10 – EIC** Interrupt Flag for EIC
This bit is set when a Peripheral Access Error occurs while accessing the EIC, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 9 – RTC** Interrupt Flag for RTC
This bit is set when a Peripheral Access Error occurs while accessing the RTC, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 8 – WDT** Interrupt Flag for WDT
This bit is set when a Peripheral Access Error occurs while accessing the WDT, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 7 – GCLK** Interrupt Flag for GCLK
This bit is set when a Peripheral Access Error occurs while accessing the GCLK, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 6 – SUPC** Interrupt Flag for SUPC
This bit is set when a Peripheral Access Error occurs while accessing the SUPC, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 5 – OSC32KCTRL** Interrupt Flag for OSC32KCTRL
This bit is set when a Peripheral Access Error occurs while accessing the OSC32KCTRL, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 4 – OSCCTRL** Interrupt Flag for OSCCTRL
This bit is set when a Peripheral Access Error occurs while accessing the OSCCTRL, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 3 – RSTC** Interrupt Flag for RSTC
This bit is set when a Peripheral Access Error occurs while accessing the RSTC, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit will clear the flag.

**Bit 2 – MCLK** Interrupt Flag for MCLK
This bit is set when a Peripheral Access Error occurs while accessing the MCLK, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 1 – PM** Interrupt Flag for PM
This bit is set when a Peripheral Access Error occurs while accessing the PM, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 0 – PAC** Interrupt Flag for PAC
This bit is set when a Peripheral Access Error occurs while accessing the PAC, and will generate an interrupt request if SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.7.7. Peripheral Interrupt Flag Status - Bridge B

**Name:** INTFLAGB
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** –

These flags are set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGx bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits will clear the corresponding INTFLAGx interrupt flag.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | RAMECC |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | TC3 | TC2 | TCC1 | TCC0 | SERCOM3 | SERCOM2 | |
| Access | | RW | RW | RW | RW | RW | RW | |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EVSYS | | DMAC | PORT | CMCC | NVMCTRL | DSU | USB |
| Access | RW | | RW | RW | RW | RW | RW | RW |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 16 – RAMECC** Interrupt Flag for RAMECC
This flag is set when a Peripheral Access Error occurs while accessing the RAMECC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the RAMECC interrupt flag.

**Bit 14 – TC3** Interrupt Flag for TC3
This flag is set when a Peripheral Access Error occurs while accessing the TC3, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TC3 interrupt flag.

**Bit 13 – TC2** Interrupt Flag for TC2
This flag is set when a Peripheral Access Error occurs while accessing the TC2, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TC2 interrupt flag.

**Bit 12 – TCC1** Interrupt Flag for TCC1
This flag is set when a Peripheral Access Error occurs while accessing the TCC1, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TCC1 interrupt flag.

**Bit 11 – TCC0** Interrupt Flag for TCC0
This flag is set when a Peripheral Access Error occurs while accessing the TCC0, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TCC0 interrupt flag.

**Bit 10 – SERCOM3** Interrupt Flag for SERCOM3
This flag is set when a Peripheral Access Error occurs while accessing the SERCOM3, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the SERCOM3 interrupt flag.

**Bit 9 – SERCOM2** Interrupt Flag for SERCOM2
This flag is set when a Peripheral Access Error occurs while accessing the SERCOM2, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the SERCOM2 interrupt flag.

**Bit 7 – EVSYS** Interrupt Flag for EVSYS
This flag is set when a Peripheral Access Error occurs while accessing the EVSYS, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the EVSYS interrupt flag.

**Bit 5 – DMAC** Interrupt Flag for DMAC
This flag is set when a Peripheral Access Error occurs while accessing the DMAC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the DMAC interrupt flag.

**Bit 4 – PORT** Interrupt Flag for PORT
This flag is set when a Peripheral Access Error occurs while accessing the PORT, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the PORT interrupt flag.

**Bit 3 – CMCC** Interrupt Flag for CMCC
This flag is set when a Peripheral Access Error occurs while accessing the CMCC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the CMCC interrupt flag.

**Bit 2 – NVMCTRL** Interrupt Flag for NVMCTRL
This flag is set when a Peripheral Access Error occurs while accessing the NVMCTRL, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the NVMCTRL interrupt flag.

**Bit 1 – DSU** Interrupt Flag for DSU
This flag is set when a Peripheral Access Error occurs while accessing the DSU, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit will clear the DSU interrupt flag.

**Bit 0 – USB** Interrupt Flag for USB

This flag is set when a Peripheral Access Error occurs while accessing the USB, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the USB interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.7.8. Peripheral Interrupt Flag Status - Bridge C

**Name:** INTFLAGC
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** –

These flags are set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGx bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits will clear the corresponding INTFLAGx interrupt flag.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | CCL | QSPI | PUKCC | ICM | TRNG | AES | AC |
| Access | | RW | RW | RW | RW | RW | RW | RW |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PDEC | TC5 | TC4 | TCC3 | TCC2 | GMAC | CAN1 | CAN0 |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 14 – CCL** Interrupt Flag for CCL
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the CCL, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the CCL interrupt flag.

**Bit 13 – QSPI** Interrupt Flag for QSPI
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the QSPI, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the QSPI interrupt flag.

**Bit 12 – PUKCC** Interrupt Flag for PUKCC
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the PUKCC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the PUKCC interrupt flag.

**Bit 11 – ICM** Interrupt Flag for ICM
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the ICM, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the ICM interrupt flag.

**Bit 10 – TRNG** Interrupt Flag for TRNG
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the TRNG, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TRNG interrupt flag.

**Bit 9 – AES** Interrupt Flag for AES
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the AES, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the AES interrupt flag.

**Bit 8 – AC** Interrupt Flag for AC
This flag is set when a Peripheral Access Error occurs while accessing the peripheral associated with the AC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the AC interrupt flag.

**Bit 7 – PDEC** Interrupt Flag for PDEC
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the PDEC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the PDEC interrupt flag.

**Bit 6 – TC5** Interrupt Flag for TC5
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the TC5, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TC5 interrupt flag.

**Bit 5 – TC4** Interrupt Flag for TC4
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the TC4, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TC4 interrupt flag.

**Bit 4 – TCC3** Interrupt Flag for TCC3
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the TCC3, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TCC3 interrupt flag.

**Bit 3 – TCC2** Interrupt Flag for TCC2
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the TCC2, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the TCC2 interrupt flag.

**Bit 2 – GMAC** Interrupt Flag for GMAC
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the GMAC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit will clear the GMAC interrupt flag.

**Bit 1 – CAN1** Interrupt Flag for CAN1
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the CAN1, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the CAN1 interrupt flag.

**Bit 0 – CAN0** Interrupt Flag for CAN0
This flags is set when a Peripheral Access Error occurs while accessing the peripheral associated with the CAN0, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the CAN0 interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 27.7.9. Peripheral Interrupt Flag Status - Bridge D

**Name:** INTFLAGD
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** –

These flags are set when a Peripheral Access Error occurs while accessing the peripheral associated with the respective INTFLAGx bit, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.

Writing a '0' to these bits has no effect.

Writing a '1' to these bits will clear the corresponding INTFLAGx interrupt flag.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PCC | I2S | DAC | ADC1 |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADC0 | TC7 | TC6 | TCC4 | SERCOM7 | SERCOM6 | SERCOM5 | SERCOM4 |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 11 – PCC** Interrupt Flag for PCC
This flag is set when a Peripheral Access Error occurs while accessing the PCC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the PCC interrupt flag.

**Bit 10 – I2S** Interrupt Flag for I2S
This flag is set when a Peripheral Access Error occurs while accessing the I2S, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the I2S interrupt flag.

**Bit 9 – DAC** Interrupt Flag for DAC
This flag is set when a Peripheral Access Error occurs while accessing the DAC, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the DAC interrupt flag.

**Bit 8 – ADC1** Interrupt Flag for ADC1
This flag is set when a Peripheral Access Error occurs while accessing the ADC1, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the ADC1 interrupt flag.

**Bit 7 – ADC0** Interrupt Flag for ADC0
This flag is set when a Peripheral Access Error occurs while accessing the ADC0, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the ADC0 interrupt flag.

**Bit 6 – TC7** Interrupt Flag for TC7
This flag is set when a Peripheral Access Error occurs while accessing the TC6, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the TC7 interrupt flag.

**Bit 5 – TC6** Interrupt Flag for TC6
This flag is set when a Peripheral Access Error occurs while accessing the TC6, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the TC6 interrupt flag.

**Bit 4 – TCC4** Interrupt Flag for TCC4
This flag is set when a Peripheral Access Error occurs while accessing the TCC4, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the TCC4 interrupt flag.

**Bit 3 – SERCOM7** Interrupt Flag for SERCOM7
This flag is set when a Peripheral Access Error occurs while accessing the SERCOM7, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the SERCOM7 interrupt flag.

**Bit 2 – SERCOM6** Interrupt Flag for SERCOM6
This flag is set when a Peripheral Access Error occurs while accessing the SERCOM6, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the SERCOM6 interrupt flag.

**Bit 1 – SERCOM5** Interrupt Flag for SERCOM5
This flag is set when a Peripheral Access Error occurs while accessing the SERCOM5, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the SERCOM5 interrupt flag.

**Bit 0 – SERCOM4** Interrupt Flag for SERCOM4
This flag is set when a Peripheral Access Error occurs while accessing the SERCOM4, and will generate an interrupt request if INTENCLR/SET.ERR is '1'.
Writing a '0' to these bits has no effect.
Writing a '1' to these bits will clear the SERCOM4 interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 27.7.10. Peripheral Write Protection Status A

**Name:** STATUSA
**Offset:** 0x34
**Reset:** 0x00010000
**Property:** PAC Write-Protection

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected. |
| 1 | Peripheral is write protected. |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TC1 | TC0 | SERCOM1 | SERCOM0 | FREQM | EIC | RTC | WDT |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GCLK | SUPC | OSC32KCTRL | OSCCTRL | RSTC | MCLK | PM | PAC |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – TC1** TC1 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TC1 is not write protected |
| 1 | TC1 is write protected |

**Bit 14 – TC0** TC0 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TC0 is not write protected |
| 1 | TC0 is write protected |

**Bit 13 – SERCOM1** SERCOM1 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM1 is not write protected |
| 1 | SERCOM1 is write protected |

**Bit 12 – SERCOM0** SERCOM0 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM0 is not write protected |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | SERCOM0 is write protected |

**Bit 11 – FREQM** FREQM APB Protect Enable

| Value | Description |
|---|---|
| 0 | FREQM is not write protected |
| 1 | FREQM is write protected |

**Bit 10 – EIC** EIC APB Protect Enable

| Value | Description |
|---|---|
| 0 | EIC is not write protected |
| 1 | EIC is write protected |

**Bit 9 – RTC** RTC APB Protect Enable

| Value | Description |
|---|---|
| 0 | RTC is not write protected |
| 1 | RTC is write protected |

**Bit 8 – WDT** WDT APB Protect Enable

| Value | Description |
|---|---|
| 0 | WDT is not write protected |
| 1 | WDT is write protected |

**Bit 7 – GCLK** GCLK APB Protect Enable

| Value | Description |
|---|---|
| 0 | GCLK is not write protected |
| 1 | GCLK is write protected |

**Bit 6 – SUPC** SUPC APB Protect Enable

| Value | Description |
|---|---|
| 0 | SUPC is not write protected |
| 1 | SUPC is write protected |

**Bit 5 – OSC32KCTRL** OSC32KCTRL APB Protect Enable

| Value | Description |
|---|---|
| 0 | OSC32KCTRL is not write protected |
| 1 | OSC32KCTRL is write protected |

**Bit 4 – OSCCTRL** OSCCTRL APB Protect Enable

| Value | Description |
|---|---|
| 0 | OSCCTRL is not write protected |
| 1 | OSCCTRL is write protected |

**Bit 3 – RSTC** RSTC APB Protect Enable

| Value | Description |
|---|---|
| 0 | RSTC is not write protected |
| 1 | RSTC is write protected |

**Bit 2 – MCLK** MCLK APB Protect Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | MCLK is not write protected |
| 1 | MCLK is write protected |

**Bit 1 – PM** PM APB Protect Enable

| Value | Description |
|---|---|
| 0 | PM is not write protected |
| 1 | PM is write protected |

**Bit 0 – PAC** PAC APB Protect Enable

| Value | Description |
|---|---|
| 0 | PAC is not write protected |
| 1 | PAC is write protected |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 27.7.11. Peripheral Write Protection Status - Bridge B

**Name:** STATUSB
**Offset:** 0x38
**Reset:** 0x00000002
**Property:** PAC Write-Protection

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected. |
| 1 | Peripheral is write protected. |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | RAMECC |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | TC3 | TC2 | TCC1 | TCC0 | SERCOM3 | SERCOM2 | |
| Access | | R | R | R | R | R | R | |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EVSYS | | DMAC | PORT | CMCC | NVMCTRL | DSU | USB |
| Access | R | | R | R | R | R | R | R |
| Reset | 0 | | 0 | 0 | 0 | 0 | 1 | 0 |

**Bit 16 – RAMECC** RAMECC APB Protect Enable

| Value | Description |
|---|---|
| 0 | RAMECC peripheral is not write protected |
| 1 | RAMECC peripheral is write protected |

**Bit 14 – TC3** TC3 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TC3 peripheral is not write protected |
| 1 | TC3 peripheral is write protected |

**Bit 13 – TC2** TC2 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TC2 peripheral is not write protected |
| 1 | TC2 peripheral is write protected |

**Bit 12 – TCC1** TCC1 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TCC1 peripheral is not write protected |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | TCC1 peripheral is write protected |

**Bit 11 – TCC0** TCC0 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TCC0 peripheral is not write protected |
| 1 | TCC0 peripheral is write protected |

**Bit 10 – SERCOM3** SERCOM3 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM3 peripheral is not write protected |
| 1 | SERCOM3 peripheral is write protected |

**Bit 9 – SERCOM2** SERCOM2 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM2 peripheral is not write protected |
| 1 | SERCOM2 peripheral is write protected |

**Bit 7 – EVSYS** EVSYS APB Protect Enable

| Value | Description |
|---|---|
| 0 | EVSYS peripheral is not write protected |
| 1 | EVSYS peripheral is write protected |

**Bit 5 – DMAC** DMAC APB Protect Enable

| Value | Description |
|---|---|
| 0 | DMAC peripheral is not write protected |
| 1 | DMAC peripheral is write protected |

**Bit 4 – PORT** PORT APB Protect Enable

| Value | Description |
|---|---|
| 0 | PORT peripheral is not write protected |
| 1 | PORT peripheral is write protected |

**Bit 3 – CMCC** CMCC APB Protect Enable

| Value | Description |
|---|---|
| 0 | CMCC peripheral is not write protected |
| 1 | CMCC peripheral is write protected |

**Bit 2 – NVMCTRL** NVMCTRL APB Protect Enable

| Value | Description |
|---|---|
| 0 | NVMCTRL peripheral is not write protected |
| 1 | NVMCTRL peripheral is write protected |

**Bit 1 – DSU** DSU APB Protect Enable

| Value | Description |
|---|---|
| 0 | DSU peripheral is not write protected |
| 1 | DSU peripheral is write protected |

**Bit 0 – USB** USB APB Protect Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | USB peripheral is not write protected |
| 1 | USB peripheral is write protected |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 27.7.12. Peripheral Write Protection Status - Bridge C

**Name:** STATUSC
**Offset:** 0x3C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected. |
| 1 | Peripheral is write protected. |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | CCL | QSPI | PUKCC | ICM | TRNG | AES | AC |
| Access | | R | R | R | R | R | R | R |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PDEC | TC5 | TC4 | TCC3 | TCC2 | GMAC | CAN1 | CAN0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 14 – CCL** CCL APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 13 – QSPI** QSPI APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 12 – PUKCC** PUKCC APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 11 – ICM** ICM APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Peripheral is write protected |

**Bit 10 – TRNG** TRNG APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 9 – AES** AES APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 8 – AC** AC APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 7 – PDEC** PDEC APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 6 – TC5** TC5 APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 5 – TC4** TC4 APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 4 – TCC3** TCC3 APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 3 – TCC2** TCC2 APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 2 – GMAC** GMAC APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 1 – CAN1** CAN1 APB Protection Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |

**Bit 0 – CAN0** CAN0 APB Protection Enable

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected |
| 1 | Peripheral is write protected |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 27.7.13. Peripheral Write Protection Status - Bridge D

**Name:** STATUSD
**Offset:** 0x40
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Read-Only

Writing to this register has no effect.

Reading STATUS register returns peripheral write protection status:

| Value | Description |
|---|---|
| 0 | Peripheral is not write protected. |
| 1 | Peripheral is write protected. |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PCC | I2S | DAC | ADC1 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADC0 | TC7 | TC6 | TCC4 | SERCOM7 | SERCOM6 | SERCOM5 | SERCOM4 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 11 – PCC** PCC APB Protect Enable

| Value | Description |
|---|---|
| 0 | PCC is not write protected |
| 1 | PCC is write protected |

**Bit 10 – I2S** I2S APB Protect Enable

| Value | Description |
|---|---|
| 0 | I2S is not write protected |
| 1 | I2S is write protected |

**Bit 9 – DAC** DAC APB Protect Enable

| Value | Description |
|---|---|
| 0 | DAC is not write protected |
| 1 | DAC is write protected |

**Bit 8 – ADC1** ADC1 APB Protect Enable

| Value | Description |
|---|---|
| 0 | ADC1 is not write protected |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | ADC1 is write protected |

**Bit 7 – ADC0** ADC0 APB Protect Enable

| Value | Description |
|---|---|
| 0 | ADC0 is not write protected |
| 1 | ADC0 is write protected |

**Bit 6 – TC7** TC7 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TC7 is not write protected |
| 1 | TC7 is write protected |

**Bit 5 – TC6** TC6 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TC6 is not write protected |
| 1 | TC6 is write protected |

**Bit 4 – TCC4** TCC4 APB Protect Enable

| Value | Description |
|---|---|
| 0 | TCC4 is not write protected |
| 1 | TCC4 is write protected |

**Bit 3 – SERCOM7** SERCOM7 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM7 is not write protected |
| 1 | SERCOM7 is write protected |

**Bit 2 – SERCOM6** SERCOM6 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM6 is not write protected |
| 1 | SERCOM6 is write protected |

**Bit 1 – SERCOM5** SERCOM5 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM5 is not write protected |
| 1 | SERCOM5 is write protected |

**Bit 0 – SERCOM4** SERCOM4 APB Protect Enable

| Value | Description |
|---|---|
| 0 | SERCOM4 is not write protected |
| 1 | SERCOM4 is write protected |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 28. OSCCTRL – Oscillators Controller

## 28.1. Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSCn, DFLL48M, and two FDPLL200Mn modules. Another clock controller (OSC32KCTRL) provides a user interface to 32.768 kHz oscillators.

Through the interface registers, it is possible to enable, disable, calibrate, and monitor the oscillators.

The status of all oscillators are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes via the INTENSET, INTENCLR, and INTFLAG registers.

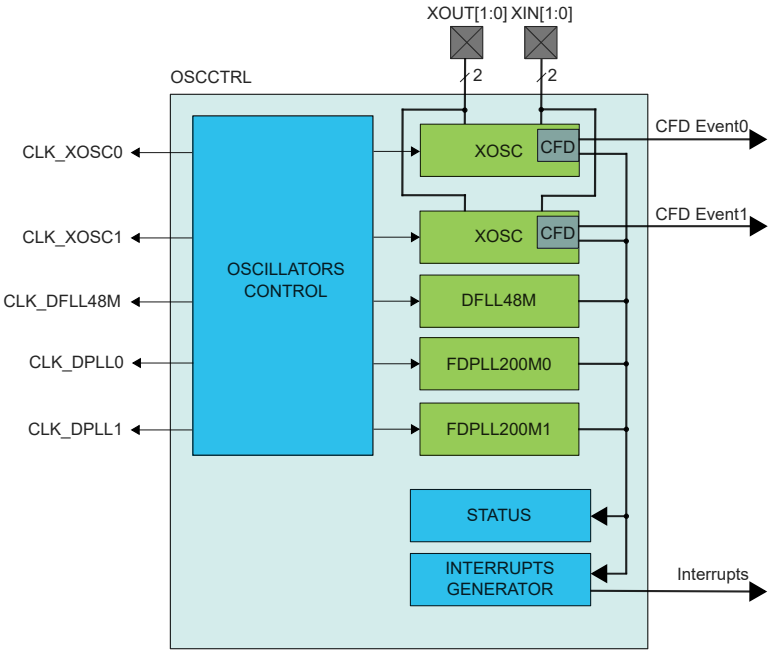
## 28.2. Features

- Digital Frequency-Locked Loop (DFLL48M)
  - Internal oscillator with no external components
  - 48 MHz output frequency
  - Operates stand-alone as a high-frequency programmable oscillator in Open Loop mode
  - Operates as an accurate frequency multiplier against a known frequency in Closed Loop mode
- Two 8-48 MHz Crystal Oscillators (XOSCn)
  - Tunable gain control
  - Programmable start-up time
  - Crystal or external input clock on XIN I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- Two Digital Phase-Locked Loop (FDPLL200Mn)
  - 96 MHz to 200 MHz output frequency from a 32 kHz to 3.2 MHz reference clock
  - Two DPLLs, each with four selectable reference clocks
  - Adjustable digital filter for jitter optimization
  - Adjustable DCO filter for a 4-stages differential ring oscillator
  - Fractional part used to achieve 1/32th of reference clock step
  - Embedded test mode controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 28.3. Block Diagram

**Figure 28-1.** OSCCTRL Block Diagram




## 28.4. Signal Description

| Signal | Description | Type |
|---|---|---|
| XIN[1:0] | Multipurpose Crystal Oscillator or external clock generator input | Analog input |
| XOUT[1:0] | Multipurpose Crystal Oscillator output | Analog output |

The I/O lines are automatically selected when XOSCn is enabled.

## 28.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 28.5.1. I/O Lines

I/O lines are configured by OSCCTRL when XOSCn is enabled, and need no user configuration.

### 28.5.2. Power Management

The OSCCTRL can continue to operate in any sleep mode where the selected source clock is running. The OSCCTRL interrupts can be used to wake up the device from sleep modes. The events can trigger other operations in the system without exiting sleep modes.

**Related Links**

PM – Power Manager

### 28.5.3. Clocks

The OSCCTRL gathers controls for all device oscillators and provides clock sources to the Generic Clock Controller (GCLK). The available clock sources are: XOSCn, DFLL48M, and FDPLL200Mn.

The DFLL48M requires a reference clock (GCLK_DFLL48M_REF) from the GCLK. The control logic uses the oscillator output, which is also asynchronous to the user interface clock (CLK_OSCCTRL_APB).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

The FDPLL200Mn require a reference clock (GCLK_DPLL) for the FDPLL output. When the optional lock timer timeout function is used, a 32KHz reference clock (GCLK_DPLL_32K) is also required. Both reference clocks can either stem from the GCLK and/or from external oscillators.

### 28.5.4. DMA

Not applicable.

### 28.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the OSCCTRL interrupts requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 28.5.6. Events

The events of this peripheral are connected to the Event System.

**Related Links**

Event System (EVSYS)

### 28.5.7. Debug Operation

When the CPU is halted in debug mode the OSCCTRL continues normal operation. If the OSCCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

### 28.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear register (INTFLAG)

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

### 28.5.9. Analog Connections

The 8-48 MHz crystal must be connected between the XIN and XOUT pins, along with any required load capacitors.

**Note:** Refer to the Electrical Characteristics for more information about load capacitors.

## 28.6. Functional Description

### 28.6.1. Principle of Operation

XOSC, DFLL48M, and FDPLL200M are configured via OSCCTRL control registers. Through this interface, these oscillators are enabled, disabled, or have their calibration values updated.

The Status register gathers different status signals coming from the oscillators controlled by the OSCCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from Sleep mode, provided the corresponding interrupt is enabled.

### 28.6.2. External Multipurpose Crystal Oscillator (XOSCn) Operation

The XOSCn can operate in two different modes:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- External clock, with an external clock signal connected to the XIN pin
- Crystal oscillator, with an external 8-48 MHz crystal connected to the XIN and XOUT pins

The XOSCn can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller (GCLK).

At reset, the XOSCn is disabled, and the XINn/XOUTn pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSCn is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XINn and XOUTn pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XINn pins will be overridden and controlled by the OSCCTRL, while the XOUTn pins can still be used as GPIO pins.

The XOSCn is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRLn.ENABLE). To enable XOSCn as an external crystal oscillator, the XTAL Enable bit (XOSCCTRLn.XTALEN) must written to '1'. If XOSCCTRLn.XTALEN is zero, the external clock input on XIN will be enabled.

When in crystal oscillator mode (XOSCCTRLn.XTALEN=1), the External Multipurpose Crystal Oscillator Current Control (XOSCCTRLn.IPTAT, XOSCCTRLn.IMULT) must be set to match the external crystal oscillator frequency. If the External Multipurpose Crystal Oscillator Enable Amplitude Loop Control (XOSCCTRLn.ENALC) is '1', the oscillator amplitude will be automatically adjusted, and in most cases result in lower power consumption.

The bias current of the Crystal Oscillator can be adjusted to the desired value for a proper oscillation by setting the bit fields XOSCCTRLn.IPTAT and XOSCCTRLn.IMULT. See the recommended setting in the Table 28-7 table.

The low buffer gain is used to adjust the oscillator's amplitude in automatic loop control (XOSCCTRLn.ENALC=1). The default value of LOWBUFGAIN=0 should be used to allow operating with a low amplitude oscillator. The setting LOWBUFGAIN=1 can be used to solve stability issues. If set, the oscillator's amplitude is increased by a factor of approximately 2.

The XOSCn will behave differently in different sleep modes, based on the settings of XOSCCTRLn.RUNSTDBY, XOSCCTRLn.ONDEMAND, and XOSCCTRLn.ENABLE

**Table 28-1.** XOSC Sleep Behavior

| XOSCCTRLn.RUNSTDBY | XOSCCTRLn.ONDEMAND | XOSCTRLn.ENABLE | Sleep Behavior |
|---|---|---|---|
| - | - | 0 | Disabled |
| 0 | 0 | 1 | Always run in Idle Sleep modes. Run in Standby Sleep mode if requested by a peripheral. |
| 0 | 1 | 1 | Only run in Idle or Standby Sleep modes if requested by a peripheral. |
| 1 | 0 | 1 | Always run in Idle and Standby Sleep modes. |
| 1 | 1 | 1 | Only run in Idle or Standby Sleep modes if requested by a peripheral. |

After a hard reset, or when waking up from a sleep mode where the XOSCn was disabled, the XOSCn will need a certain amount of time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSCCTRLn.STARTUP) in the External Multipurpose Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic. The External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDYn) is set when the external clock or crystal oscillator is stable and ready to be used as a clock source. An interrupt is generated on a zero-to-one transition on STATUS.XOSCRDYn if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDYn) is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
GCLK - Generic Clock Controller

### 28.6.3. Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator clock signal provided by the External Multipurpose Crystal Oscillator (XOSCn). It detects failing operation of the XOSCn clock, and supports switching to a safe clock in case of clock failure. The user can also switch from the safe clock to the XOSCn clock in case of clock recovery. The safe clock is derived from the DFLL48M with a configurable prescaler. This supports configuring the safe clock in order to fulfill the operative conditions of the microcontroller. The CFD operation is automatically suspended when the XOSCn clock is not requested in ONDEMAND mode or halted in STANDBY.

The user interface registers allow to enable, disable and configure the CFD. The Status register gives status on failure and clock switch conditions. The Clock Failure Detector can optionally trigger an interrupt or an event when a failure is detected.

**Clock Failure Detection**

At reset, the CFD is disabled. The CFD does not monitor the XOSCn clock when the oscillator is disabled (XOSCCTRLn.ENABLE = 0).

Before starting the CFD operation, the user must start and enable the safe clock source (DFLL48M). To start the CFD operation, the user must write a one to the CFD Enable bit in the External Oscillator Control register (XOSCCTRLn.CFDEN). After the start or restart of the XOSCn, the CFD does not detect failure until the start-up time, as configured by the Oscillator Start-Up Time (XOSCCTRLn.STARTUP) in the External Multipurpose Crystal Oscillator Control register, is elapsed. Once the XOSCn Start-Up Time is elapsed, the XOSCn clock is constantly monitored.

During a period of 4 safe clocks , the CFD watches for a clock activity from the XOSCn. There must be one rising and one falling XOSCn clock edges during a 4 safe clock periods to meet a non failure status. If no activity is detected, the failure status is asserted. The Clock Failure status bit in the Status register (STATUS.XOSCFAILn) is set. The Clock Failure interrupt flag bit in the Interrupt Flag register (INTFLAG.XOSCFAILn) is set. If the XOSCFAILn bit in the Interrupt Enable Set register (INTENSET.XOSCFAILn) is set, an interrupt is generated . An output event is generated as well, if the Event Output enable bit in the Event Control register (EVCTRL.CFDEOn) is set.

The XOSCn clock continues to be monitored after a clock failure. The Clock Failure status bit in the Status register (STATUS.XOSCFAILn) reflects the current XOSCn clock activity.

**Clock Switch**

When a clock failure is detected, the XOSCn clock is replaced by the safe clock in order to maintain an active clock during the XOSCn clock failure. The safe clock source is the DFLL48M oscillator clock. The safe clock source can be downscaled with a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSCn clock is switched to the safe clock, the Clock Switch bit (STATUS.XOSCCKSWn) in the Status register is set.

When the CFD has switched to the safe clock, the XOSCn is not disabled. The application must take the necessary actions to disable the oscillator N. The application must also take the necessary actions to configure the system clocks to continue normal operations.

In the case the application can recover the XOSCn , it can switch back to the XOSCn clock by writing a one to Switch Enable bit (XOSCCTRLn.SWBEN) in the External Oscillator Control register. Once the XOSCn clock is switched back, the Switch Back bit (XOSCCTRLn.SWBEN) is cleared by the hardware.

**Prescaler**

The CFD has an internal configurable prescaler (XOSCCTRLn.CFDPRESC) to generate the safe clock from the DFLL48M clock. The prescaler size allows to scale down the DFLL48M clock such that


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

the safe clock is not higher than the XOSCn clock frequency monitored by the CFD. The frequency divider is 2^CFDPRESC where CFDPRESC range from 0 to 15.

Example: for an external crystal oscillator at 8 mHz and the DFLL48M internal oscillator configured to generate a 48 mHz clock, the prescaler should select a downscale value above 6 (48/8), eg. 8, thus CFDPRESC=3.

#### Event

If the Event Output enable bit in the Event Control register (EVCTRL.CFDEOn) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

#### Sleep Mode

The CFD is halted depending on configuration of the XOSCn and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

### 28.6.4. Digital Frequency Locked Loop (DFLL48M) Operation

The DFLL48M can operate in both open-loop mode and closed-loop mode. In closed-loop mode, a low-frequency clock with high accuracy should be used as the DFLL48M reference clock to get high accuracy on the output clock (CLK_DFLL48M).

The DFLL48M's output can be used as a source for the generic clock generators (GCLK.GENCTRLn.SRC = DFLL).

#### Related Links

GCLK - Generic Clock Controller

#### 28.6.4.1. Basic Operation

##### Operating modes

The DFLL48M will behave differently in different sleep modes based on the settings of DFLLCTRLA.RUNSTDBY, DFLLCTRLA.ONDEMAND and DFLLCTRLA.ENABLE, as shown in the following table.

**Table 28-2.** DFLL48M Sleep Behavior

| DFLLCTRLA.RUNSTDBY | DFLLCTRLA.ONDEMAND | DFLLCTRLA.ENABLE | Sleep Behavior |
|---|---|---|---|
| - | - | 0 | Disabled |
| 0 | 0 | 1 | Always run in Idle Sleep mode. Run in Standby Sleep mode if requested by a peripheral. |
| 0 | 1 | 1 | Only run in Idle or Standby Sleep modes if requested by a peripheral. |
| 1 | 0 | 1 | Always run in Idle and Standby Sleep modes. |
| 1 | 1 | 1 | Only run in Idle or Standby Sleep mode if requested by a peripheral. |

The DFLL48M is used as a clock source for the generic clock generators, as described in the GCLK chapter.

The DFLL48M is factory-calibrated to provide 48 MHz in open-loop mode. The DFLLVAL.COARSE and DFLLVAL.FINE registers store the factory frequency calibration values after reset.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### Open-Loop Operation

After any reset, the open-loop mode is selected. When operating in open-loop mode, the output frequency of the DFLL48M will be determined by the values written to the DFLL Coarse Value bit group and the DFLL Fine Value bit group (DFLLVAL.COARSE and DFLLVAL.FINE) in the DFLL Value register.

It is possible to change the values of DFLLVAL.COARSE and DFLLVAL.FINE and thereby the output frequency of the DFLL48M output clock, CLK_DFLL48M, while the DFLL48M is enabled and in use. CLK_DFLL48M is ready to be used when STATUS.DFLLRDY is set after enabling the DFLL48M.

### Closed-Loop Operation

In closed-loop operation, the output frequency is continuously regulated against a reference clock. Once the multiplication factor is set, the oscillator fine tuning is automatically adjusted. The DFLL48M must be correctly configured before closed-loop operation can be enabled. After enabling the DFLL48M, it must be configured in the following way:

1. The output of Peripheral Channel 0 provides the input reference clock to DFLL48M (GCLK_DFLL48M_REF). Set GCLK.PCHCTRL0.GEN to the desired Generic Clock Generator number (for example, 0 for GCLK0 and 1 for GCLK1) to select its output as the input to this peripheral channel. Select the clock source for the chosen GCLK generator via its corresponding GCLK.GENCTRLn.SRC register. Select the clock input to GCLK0 via GCLK.GENCTRL0.SRC
2. Select the maximum step size allowed in finding the coarse and fine values by writing the appropriate values to the DFLL Coarse Maximum Step and DFLL Fine Maximum Step bit groups (DFLLMUL.CSTEP and DFLLMUL. FSTEP) in the DFLL Multiplier register. A small step size will ensure low overshoot on the output frequency, but will typically result in longer lock times. A high value might give a large overshoot, but will typically provide faster locking. DFLLMUL.CSTEP and DFLLMUL.FSTEP should not be higher than 50% of the maximum value of DFLLVAL.COARSE and DFLLVAL.FINE.
3. Select the multiplication factor in the DFLL Multiply Factor bit group (DFLLMUL.MUL) in the DFLL Multiplier register. Care must be taken when choosing DFLLMUL.MUL so that the output frequency does not exceed the maximum frequency of the device.
4. Start the closed-loop mode by writing a one to the DFLL Mode Selection bit (DFLLCTRLB.MODE) in the DFLL Control register.

The frequency of CLK_DFLL48M (Fclkdfll48m) is given by:

$$F_{\mathrm{clkdfll48m}} = \mathrm{DFLLMUL.MUL} \times F_{\mathrm{clkdfll48mref}}$$

Where, Fclkdfll48mref is the frequency of the reference clock (CLK_DFLL48M_REF).

DFLLVAL.COARSE and DFLLVAL.FINE are read-only in closed-loop mode, and are controlled by the frequency tuner to meet user specified frequency. In closed-loop mode, the value in DFLLVAL.COARSE is used by the frequency tuner as a starting point for coarse. Writing DFLLVAL.COARSE to a value close to the final value before entering closed-loop mode will reduce the time needed to get a lock on Coarse.

### Frequency Locking

The locking of the frequency in closed-loop mode is divided into two stages. In the first, coarse stage, the control logic quickly finds the correct value for DFLLVAL.COARSE and sets the output frequency to a value close to the correct frequency. On coarse lock, the DFLL Locked on Coarse Value bit (STATUS.DFLLLCKC) in the Status register will be set.

In the second, fine stage, the control logic tunes the value in DFLLVAL.FINE so that the output frequency is very close to the desired frequency. On fine lock, the DFLL Locked on Fine Value bit (STATUS.DFLLLCKF) in the Status register will be set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

If the the ByPass Lock bit (DFLLCTRLB.BPLCKC) in the DFLL Control register is set, the coarse stage is by-passed, the DFLLVAL.COARSE keeps it's value and the DFLL Coarse Value bit (STATUS.DFLLLCKC) is immediately set.

Interrupts are generated by both STATUS.DFLLLCKC and STATUS.DFLLLCKF if INTENSET.DFLLCKC or INTENSET.DFLLCKF are written to '1'.

CLK_DFLL48M is ready to be used when the DFLL Ready bit (STATUS.DFLLRDY) in the Status register is set, but the accuracy of the output frequency depends on which locks are set. For lock times, refer to the Electrical Characteristics.

#### Frequency Error Measurement

The ratio between CLK_DFLL48M_REF and CLK48M_DFLL is measured automatically when the DFLL48M is in closed-loop mode. The difference between this ratio and the value in DFLLMUL.MUL is stored in the DFLL Multiplication Ratio Difference bit group (DFLLVAL.DIFF) in the DFLL Value register. The relative error on CLK_DFLL48M compared to the target frequency is calculated as follows:

$$\mathrm{ERROR} = \frac{\mathrm{DIFF}}{\mathrm{MUL}}$$

#### Drift Compensation

If the Stable DFLL Frequency bit (DFLLCTRLB.STABLE) in the DFLL Control register is zero, the frequency tuner will automatically compensate for drift in the CLK_DFLL48M without losing either of the locks. This means that DFLLVAL.FINE can change after every measurement of CLK_DFLL48M. If the DFLLVAL.FINE value overflows or underflows due to large drift in temperature and/or voltage, the DFLL Out Of Bounds bit (STATUS.DFLLOOB) in the Status register will be set. After an Out of Bounds error condition, the user must rewrite DFLLMUL.MUL to ensure correct CLK_DFLL48M frequency. An interrupt is generated on a zero-to-one transition on STATUS.DFLLOOB if the DFLL Out Of Bounds bit (INTENSET.DFLLOOB) in the Interrupt Enable Set register is set. This interrupt will also be set if the tuner is not able to lock on the correct Coarse value. If the Stable DFLL Frequency bit (DFLLCTRLB.STABLE) in the DFLL Control register is one, the DFLLVAL.COARSE and DFLLVAL.FINE values will stay constant after the lock. The user can check for a possible drift by reading the frequency error in the DFLL Multiplication Ratio Difference bit group (DFLLVAL.DIFF).

#### Reference Clock Stop Detection

If CLK_DFLL48M_REF stops or is running at a very low frequency (slower than CLK_DFLL48M/(2 * MULMAX)), the DFLL Reference Clock Stopped bit (STATUS.DFLLRCS) in the Status register will be set. Detecting a stopped reference clock can take a long time, on the order of $2^{17}$ CLK_DFLL48M cycles. When the reference clock is stopped, the DFLL48M will operate as if in open-loop mode. Closed-loop mode operation will automatically resume if the CLK_DFLL48M_REF is restarted. An interrupt is generated on a zero-to-one transition on STATUS.DFLLRCS if the DFLL Reference Clock Stopped bit (INTENSET.DFLLRCS) in the Interrupt Enable Set register is set.

#### Related Links

NVM User Row Mapping (UROW)
GCLK - Generic Clock Controller

### 28.6.4.2. Additional Features

#### Dealing with Delay in the DFLL in Closed-Loop Mode

The time from selecting a new CLK_DFLL48M frequency until this frequency is output by the DFLL48M can be up to several microseconds. If the value in DFLLMUL.MUL is small, this can lead to instability in the DFLL48M locking mechanism, which can prevent the DFLL48M from achieving locks. To avoid this, a chill cycle, during which the CLK_DFLL48M frequency is not measured, can be enabled. The chill cycle is enabled by default, but can be disabled by writing a one to the DFLL Chill Cycle Disable bit (DFLLCTRLB.CCDIS) in the DFLL Control register. Enabling chill cycles might double the lock time.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Another solution to this problem consists of using less strict lock requirements. This is called Quick Lock (QL), which is also enabled by default, but it can be disabled by writing a one to the Quick Lock Disable bit (DFLLCTRLB.QLDIS) in the DFLL Control register. The Quick Lock might lead to a larger spread in the output frequency than chill cycles, but the average output frequency is the same.

**Note:** During a maximum 30 cycles of the reference clock period, between lock flag asserted and frequency stabilization, DFLL accuracy will be limited. To minimize the cycle-to-cycle jitter during lock search, in addition to selecting a low step value (DFLLMUL.FSTEP, DFLLMUL.CSTEP), it is recommended to disable the Quick Lock feature.

### USB Clock Recovery Mode

USB Clock Recovery mode can be used to create the 48 MHz USB clock from the USB Start Of Frame (SOF). This mode is enabled by writing a '1' to both the USB Clock Recovery Mode bit and the Mode bit in the DFLL Control register (DFLLCTRLB.USBCRM and DFLLCTRLB.MODE).

**Note:** The DFLLCTRLB.USBCRM must not be set if USB is disabled.

Changing the reference clock source (CLK_DFLL_REF) must be done as follows:

- Disable the USB
- Switch the DFLL48M in open loop mode
- Change the reference clock (DFLLCTRLB.USBCRM)
- Switch the DFLL48M in close loop mode
- Enable the USB

**Note:** When USB Clock Recovery mode is enabled, the GCLK DFLL48M reference clock (GCLK_DFLL48M_REF) must be disabled.

In USB Clock Recovery mode, the status bits of the DFLL in OSCCTRL.STATUS are determined by the USB bus activity, and have no valid meaning. The SOF signal from USB device will be used as reference clock (CLK_DFLL_REF), ignoring the selected generic clock reference. When the USB device is connected, a SOF will be sent every 1ms, thus DFLLVAL.MUX bits should be written to 0xBB80 to obtain a 48MHz clock. In USB clock recovery mode, the DFLLCTRLB.BPLCKC bit state is ignored, and the value stored in the DFLLVAL.COARSE will be used as final Coarse value.

The COARSE value for a calibrated 48 MHz frequency is loaded from NVM after any system reset and may vary in operating modes different of the USB Clock Recovery Mode. The initial COARSE value can be saved and restored by the software if necessary.

The locking procedure will also go instantaneously to the fine lock search.

The DFLLCTRLB.QLDIS bit must be cleared and DFLLCTRLB.CCDIS should be set to speed up the lock phase. The DFLLCTRLB.STABLE bit state is ignored, an auto jitter reduction mechanism is used instead.

### Wake from Sleep Modes

DFLL48M can optionally reset its lock bits when it is disabled. This is configured by the Lose Lock After Wake bit (DFLLCTRLB.LLAW) in the DFLL Control register. If DFLLCTRLB.LLAW is zero, the DFLL48M will be re-enabled and start running with the same configuration as before being disabled, even if the reference clock is not available. The locks will not be lost. Thus it is important that the user checks that the DFLL48M has reached the COARSE and FINE lock stage before entering a sleep mode. When the reference clock has restarted, the Fine tracking will quickly compensate for any frequency drift during sleep if DFLLCTRLB.STABLE is zero. If DFLLCTRLB.LLAW is one when disabling the DFLL48M, the DFLL48M will lose all its locks, and needs to regain these through the full lock sequence.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### Wait for Lock

DFLL48M can optionally control the issued clock. This is configured by the Wait For Lock bit (DFLLCTRLB.WAITLOCK) in the DFLL Control register. If DFLLCTRLB.WAITLOCK is zero, the DFLL48M will issue a clock immediately after the ready bit (STATUS.DFLLRDY) has risen. If DFLLCTRLB.WAITLOCK is one, the DFLL48M will issue a clock immediately after the fine lock bit (STATUS.DFLLCKF) has risen. Using the wait for lock feature allows a better accuracy of the issued DFLL48M clock, conversely it increases the startup time of the DFLL48M clock.

### Accuracy

There are two main factors that determine the accuracy of Fclkdfll48m. These can be tuned to obtain maximum accuracy when fine lock is achieved.

- Fine resolution: The frequency step between two Fine values.
- The accuracy of the reference clock.

## 28.6.5. Digital Phase Locked Loop (FDPLL200Mn) Operation

The task of the DPLL is to maintain coherence between the input (reference) signal and the respective output frequency CLK_DPLL through phase comparison. The DPLL controller supports four independent sources of reference clocks:

- XOSC32K: This clock is provided by the 32K External Crystal Oscillator (XOSC32K).
- XOSC0 and XOSC1: These clocks are provided by the External Multipurpose Crystal Oscillator (XOSC).
- GCLK: This clock is provided by the Generic Clock Controller.

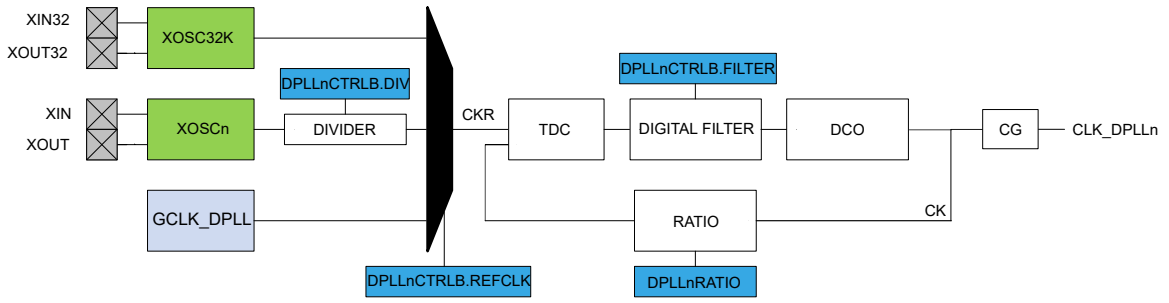
When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is as shown below:

$$f_{CLK_DPLLn} = f_{CKR} \times \left(LDR + 1 + \frac{LDRFRAC}{32}\right)$$

Where:

$f_{CLK_DPLLn}$ is the frequency of the DPLLn output clock, LDR is the loop divider ratio integer part and LDRFRAC is the loop divider ratio fractional part, $f_{CKR}$ is the frequency of the selected reference clock.

**Figure 28-2.** DPLLn Block Diagram




When the controller is disabled, the output clock is low. If the Loop Divider Ratio Fractional part bit field in the DPLL Ratio register (DPLLnRATIO.LDRFRAC) is zero, the DPLL works in Integer mode. Otherwise, the fractional mode is activated. The fractional part has a negative impact on the jitter of the DPLL.

For example (Integer mode only): Assuming $f_{CKR}$ = 32 kHz and $f_{CLK_DPLLn}$ = 112 MHz, the multiplication ratio is 3500. It means that LDR must be set to 3499.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For example (Fractional mode): Assuming $f_{CKR}$ = 32 kHz and $f_{CLK_DPPLn}$ = 112.003000 MHz, the multiplication ratio is 3500.9375 (3500 + 3/32). Thus LDR is set to 3499 and LDRFRAC to 3.

**Related Links**

GCLK - Generic Clock Controller

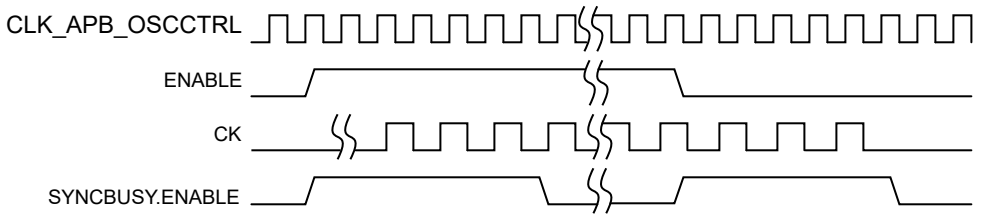
OSC32KCTRL – 32KHz Oscillators Controller

### 28.6.5.1. Basic Operation

#### Initialization, Enabling, Disabling, and Resetting

The DPLLn is enabled by writing a '1' to the Enable bit in the Control register (DPLLnCTRLA.ENABLE). The DPLLn is disabled by writing a '0' to DPLLnCTRLA.ENABLE. The DPLLnSYNCBUSY.ENABLE is set when the DPLLnCTRLA.ENABLE bit is modified. It is cleared when the DPLLn output clock CLK_DPLLn has sampled the bit at the high level, or cleared when the output clock is no longer running (for disable operation).

**Figure 28-3.** Enable Synchronization Busy Operation




The frequency of the DPLLn output clock CLK_DPLLn is stable when the module is enabled and when the LOCK bit is set. When DPLLnCTRLB.LTIME is different from 0, a user defined lock time is used to validate the lock operation. In that case the lock time is constant. If DPLLnCTRLB.LTIME is zero, the lock signal is linked with the status bit of the DPLLn (DPLLnSTATUS.LOCK), the lock time vary depending on the filter selection and final target frequency. When DPLLnCTRLB.WUF is set the wake up fast mode is activated. In that mode the clock gating cell is enabled at the end of the startup time. At that time the final frequency is not stable as it is still the acquisition period, but it allows to save hundreds of microseconds. After the first acquisition, DPLLnCTRLB.LBYPASS indicates if the Lock signal is discarded from the control of the clock gater (CG) generating the output clock CLK_DPLLn.

**Table 28-3.** CLK_DPLLn Behavior from Startup to First Edge Detection.

| WUF | LTIME | CLK_DPLLn Behavior |
|---|---|---|
| 0 | 0 | Normal Mode: First Edge when lock is asserted |
| 0 | Not Equal To Zero | Lock Timer Timeout mode: First Edge when the timer down-counts to 0. |
| 1 | X | Wake Up Fast Mode: First Edge when CK is active (startup time) |

**Table 28-4.** CLK_DPLLn Behavior after First Edge Detection.

| LBYPASS | CLK_DPLLn Behavior |
|---|---|
| 0 | Normal Mode: the CLK_DPLLn is turned off when lock signal is low. |
| 1 | Lock Bypass Mode: the CLK_DPLLn is always running, lock is irrelevant. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 28-4.** CK and CLK_DPLL Output from DPLL Off Mode to Running Mode

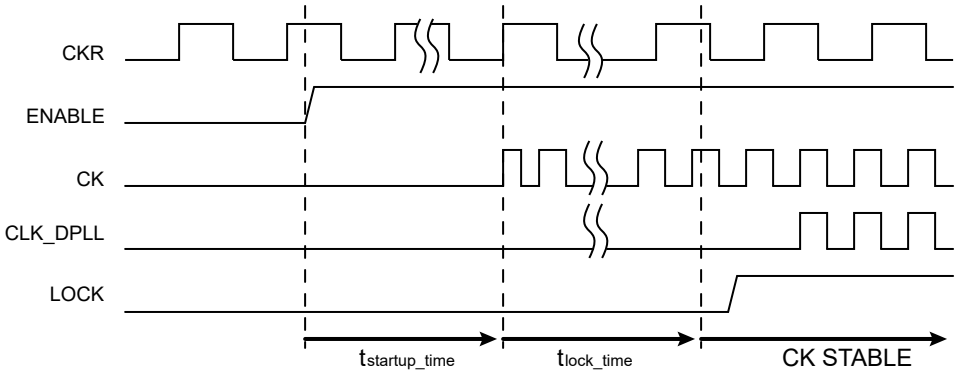



**Figure 28-5.** CK and CLK_DPLL Output from DPLL Off Mode to Running Mode when Wakeup Fast is Activated

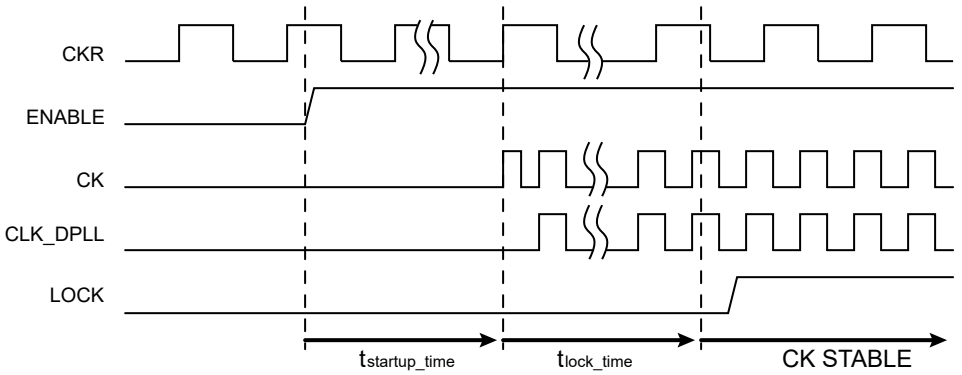
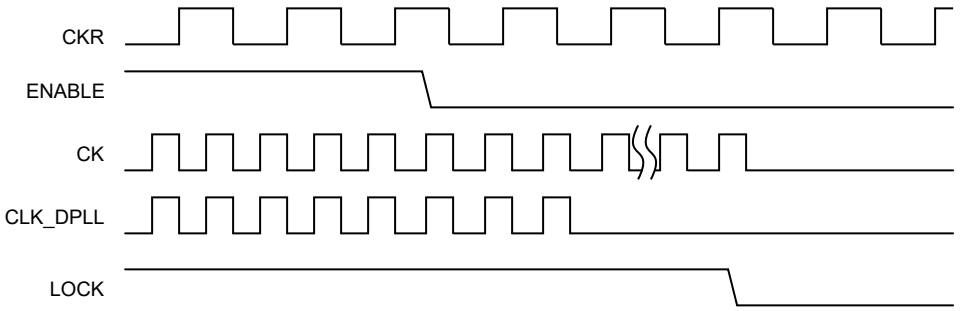



**Figure 28-6.** CK and CLK_DPLL Output from Running Mode to DPLLC Off Mode.




## Operating Modes

The DPLLn will behave differently in different sleep modes based on the settings of DPLLnCTRLA.RUNSTDBY, DPLLnCTRLA.ONDEMAND and DPLLnCTRLA.ENABLE.

**Table 28-5.** DPLL Sleep Behavior

| DPLLCTRLA.RUNSTDBY | DPLLCTRLA.ONDEMAND | DPLLCTRLA.ENABLE | Sleep Behavior |
|---|---|---|---|
| - | - | 0 | Disabled |
| 0 | 0 | 1 | Always run in Idle Sleep modes. Run in Standby Sleep mode if requested by a peripheral. |
| 0 | 1 | 1 | Only run in Idle or Standby Sleep modes if requested by a peripheral. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 28-5.** DPLL Sleep Behavior (continued)

| DPLLCTRLA.RUNSTDBY | DPLLCTRLA.ONDEMAND | DPLLCTRLA.ENABLE | Sleep Behavior |
|---|---|---|---|
| 1 | 0 | 1 | Always run in Idle and Standby Sleep modes. |
| 1 | 1 | 1 | Only run in Idle or Standby Sleep modes if requested by a peripheral. |

### Reference Clock Switching

When a software operation requires reference clock switching, the normal operation is to disable the DPLLn, modify the DPLLnCTRLB.REFCLK to select the desired reference source and activate the DPLLn again. The CLK_DPLLn output clock is ready when DPLLnSTATUS.CLKRDY bit is set.

### XOSC Reference Clock Divider

DPLLnCTRLB.DIV[10:0] bits are used to set the XOSC clock division factor and can be calculated with following formula:

$$f_{DIV} = \frac{fXOSC}{2 \times (DIV + 1)}$$

For more information, refer to DPLLnCTRLB.

### Loop Divider Ratio Updates

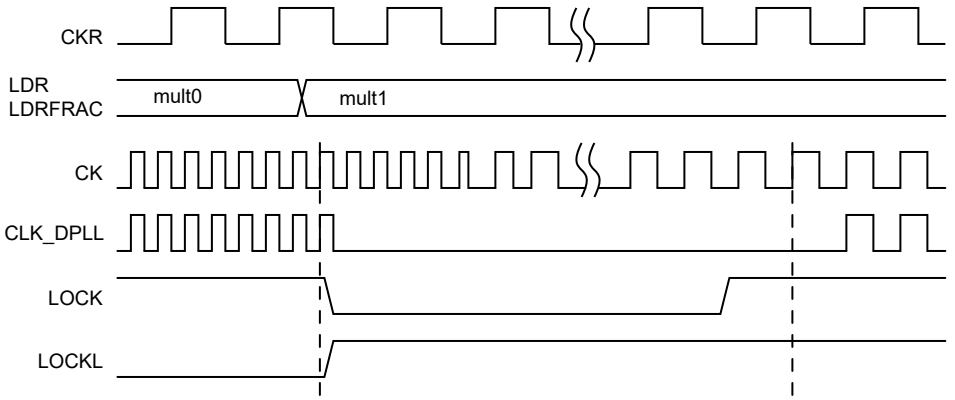
The DPLLn Controller supports on-the-fly update of the DPLLnRATIO register, so it is allowed to modify the loop divider ratio and the loop divider ratio fractional part when the DPLLn is enabled. Ensure the following conditions, or else the on-the-fly updating of the divider ratio will fail:

- DPLLnCTRLB.LBYPASS must be '0' (normal mode).
- DPLLnCTRLB.LTIME must not be 0x0, which is the default value.
- A DPLLn 32KHz clock (GCLK_DPLLn_32K) is configured in the GCLK peripheral as the internal lock timer.

Write DPLLnRATIO.LDR[12:0] bits to set the integer part of the frequency multiplier, and write DPLLnRATIO.LDRFRAC[4:0] bits to set the fractional part of the frequency multiplier. Due to synchronization there is a delay between writing to DPLLnRATIO.LDRFRAC[4:0] or DPLLnRATIO.LDR[12:0] and the effect on the DPLLn output clock. The value written DPLLnRATIO.LDRFAC[4:0] or DPLLnRATIO.LDR[12:0] will be read back immediately, and the DPLLRATIO bit in the synchronization busy register DPLLnSYNCBUSY.DPLLRATIO, will be set. DPLLnSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.STATUS.DPLLnLDRTO is set when the DPLLnRATIO register has been modified and the DPLLn analog cell has successfully sampled the updated value. At that time the DPLLnSTATUS.LOCK bit is cleared and set again by hardware when the output frequency reached a stable state. Note that if only the fractional part of loop divider ratio (DPLLnRATIO.LDRFRAC) is updated, the lock status (DPLLnSTATUS.LOCK) will not be cleared.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 28-7.** RATIOCTRL Register Update Operation




### Digital Filter Selection

The digital filter selection can be changed from the filter selection register DPLLnCTRLB.FILTER. The DPLL digital filter coefficients are automatically adjusted in order to provide a good compromise between stability and jitter. For more information, refer to DPLLnCTRLB.

### Sigma-Delta DCO Filter Selection

The sigma-delta DAC low pass filter can be controlled and adjusted from the DCO filter selection register DPLLnCTRLB.DCOFILTER[2:0]. For more information, refer to DPLLnCTRLB.

### Related Links

GCLK - Generic Clock Controller

## 28.6.6. DMA Operation

Not applicable.

## 28.6.7. Interrupts

The OSCCTRL has the following interrupt sources:

- XOSCRDYn - Multipurpose Crystal Oscillator n Ready: A 0-to-1” transition on the STATUS.XOSCRDY bit is detected
- XOSCFAILn - Clock Failure . A “0-to-1” transition on the STATUS.XOSCFAILn bit is detected
- DFLLRDY - DFLL48m Ready: A “0-to-1” transition on the STATUS.DFLLRDY bit is detected
- DFLLOOB - DFLL Out of Bounds: A "0-to-1" transition on STATUS.DFLLOOB detected
- DFLLCKF - DFLL Lock Fine: A "0-to-1" transition on STATUS.DFLLCKF detected
- DFLLCKC - DFLL Lock Coarse: A "0-to-1" transition on STATUS.DFLLCKC detected
- DPLLnLCKR - DPLLn Lock Rise: A “0-to-1” transition on the STATUS.DPLLnLCKR bit is detected
- DPLLnLCKF - DPLLn Lock Fall: A “0-to-1” transition on the STATUS.DPLLnLCKF bit is detected
- DPLLnLTTO - DPLLn Lock Timer Time-out: A “0-to-1” transition on the STATUS.DPLLnLTTO bit is detected
- DPLLnLDRTO - DPLLn Loop Divider Ratio Update Complete. A “0-to-1” transition on the STATUS.DPLLnLDRTO bit is detected

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

cleared, the interrupt is disabled, or the OSCCTRL is reset. INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

### 28.6.8. Events

The Clock Failure Detector (CFD) can generate the following output event:

- Clock Failure (XOSCFAILn): Generated when the Clock Failure status bit is set in the Status register (STATUS.XOSCFAILn). The CFD event is not generated when a Clock Switch bit (STATUS.XOSCCKSWn) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEOn) enables the CFD output event for XOSCn. Writing a '0' to this bit disables the CFD output event. Refer to the *Event System* chapter for details on configuring the event system.

### 28.6.9. Synchronization

Due to the multiple clock domains, some registers in the DFLL48M must be synchronized when accessed. A register can require:

- Synchronization when written
- Synchronization when read
- No synchronization

When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DFLLSYNC) will be set immediately, and cleared when synchronization is complete.

The following registers need synchronization:

- ENABLE bit in DFLLCTRLA register - write-synchronized
- DFLLCTRLB register - read-synchronized
- DFLLVAL register - read- and write-synchronized
- DFLLMUL register - write-synchronized

Due to the multiple clock domains (XOSC32K, XOSC, GCLK and CK), some registers in the DPLL must be synchronized when accessed. A register can require:

- Synchronization when written
- No synchronization

When executing an operation that requires synchronization, the relevant synchronization bit in the Synchronization Busy register (DPLLnSYNCBUSY) will be set immediately, and cleared when synchronization is complete.

The following bits need synchronization when written:

- Enable bit in control register A (DPLLnCTRLA.ENABLE)
- DPLLn Ratio register (DPLLnRATIO)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 28.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | EVCTRL | 7:0 | | | | | | | CFDEO1 | CFDEO0 |
| 0x01 ... 0x03 | Reserved | | | | | | | | | |
| 0x04 | INTENCLR | 7:0 | | | | | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| | | 15:8 | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| | | 23:16 | | | | | DPLL0LDRTO | DPLL0LTO | DPLL0LCKF | DPLL0LCKR |
| | | 31:24 | | | | | DPLL1LDRTO | DPLL1LTO | DPLL1LCKF | DPLL1LCKR |
| 0x08 | INTENSET | 7:0 | | | | | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| | | 15:8 | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| | | 23:16 | | | | | DPLL0LDRTO | DPLL0LTO | DPLL0LCKF | DPLL0LCKR |
| | | 31:24 | | | | | DPLL1LDRTO | DPLL1LTO | DPLL1LCKF | DPLL1LCKR |
| 0x0C | INTFLAG | 7:0 | | | | | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| | | 15:8 | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| | | 23:16 | | | | | DPLL0LDRTO | DPLL0LTO | DPLL0LCKF | DPLL0LCKR |
| | | 31:24 | | | | | DPLL1LDRTO | DPLL1LTO | DPLL1LCKF | DPLL1LCKR |
| 0x10 | STATUS | 7:0 | | | XOSCCKSW1 | XOSCCKSW0 | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| | | 15:8 | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| | | 23:16 | | | | | DPLL0LDRTO | DPLL0TO | DPLL0LCKF | DPLL0LCKR |
| | | 31:24 | | | | | DPLL1LDRTO | DPLL1TO | DPLL1LCKF | DPLL1LCKR |
| 0x14 | XOSCCTRL0 | 7:0 | ONDEMAND | RUNSTDBY | | | | XTALEN | ENABLE | |
| | | 15:8 | ENALC | IMULT[3:0] | | | | IPTAT[1:0] | | LOWBUFGAIN |
| | | 23:16 | STARTUP[3:0] | | | | | | SWBEN | CFDEN |
| | | 31:24 | | | | | CFDPRESC[3:0] | | | |
| 0x18 | XOSCCTRL1 | 7:0 | ONDEMAND | RUNSTDBY | | | | XTALEN | ENABLE | |
| | | 15:8 | ENALC | IMULT[3:0] | | | | IPTAT[1:0] | | LOWBUFGAIN |
| | | 23:16 | STARTUP[3:0] | | | | | | SWBEN | CFDEN |
| | | 31:24 | | | | | CFDPRESC[3:0] | | | |
| 0x1C | DFLLCTRLA | 7:0 | ONDEMAND | RUNSTDBY | | | | | ENABLE | |
| 0x1D ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | DFLLCTRLB | 7:0 | WAITLOCK | BPLCKC | QLDIS | CCDIS | USBCRM | LLAW | STABLE | MODE |
| 0x21 ... 0x23 | Reserved | | | | | | | | | |
| 0x24 | DFLLVAL | 7:0 | FINE[7:0] | | | | | | | |
| | | 15:8 | COARSE[5:0] | | | | | | | |
| | | 23:16 | DIFF[7:0] | | | | | | | |
| | | 31:24 | DIFF[15:8] | | | | | | | |
| 0x28 | DFLLMUL | 7:0 | MUL[7:0] | | | | | | | |
| | | 15:8 | MUL[15:8] | | | | | | | |
| | | 23:16 | FSTEP[7:0] | | | | | | | |
| | | 31:24 | CSTEP[5:0] | | | | | | | |
| 0x2C | DFLLSYNC | 7:0 | | | | DFLLMUL | DFLLVAL | DFLLCTRLB | ENABLE | |
| 0x2D ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DPLL0CTRLA | 7:0 | ONDEMAND | RUNSTDBY | | | | | ENABLE | |
| 0x31 ... 0x33 | Reserved | | | | | | | | | |
| 0x34 | DPLL0RATIO | 7:0 | LDR[7:0] | | | | | | | |
| | | 15:8 | | | | LDR[12:8] | | | | |
| | | 23:16 | | | | LDRFRAC[4:0] | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x38 | DPLL0CTRLB | 7:0 | | REFCLK[2:0] | | WUF | | FILTER[3:0] | | |
| | | 15:8 | DCOEN | DCOFILTER[2:0] | | | LBYPASS | LTIME[2:0] | | |
| | | 23:16 | DIV[7:0] | | | | | | | |
| | | 31:24 | | | | | | DIV[10:8] | | |
| 0x3C | DPLL0SYNCBUSY | 7:0 | | | | | | DPLLRATIO | ENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x40 | DPLL0STATUS | 7:0 | | | | | | | CLKRDY | LOCK |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x44 | DPLL1CTRLA | 7:0 | ONDEMAND | RUNSTDBY | | | | | ENABLE | |
| 0x45 ... 0x47 | Reserved | | | | | | | | | |
| 0x48 | DPLL1RATIO | 7:0 | LDR[7:0] | | | | | | | |
| | | 15:8 | | | | LDR[12:8] | | | | |
| | | 23:16 | | | | LDRFRAC[4:0] | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C | DPLL1CTRLB | 7:0 | | REFCLK[2:0] | | WUF | | FILTER[3:0] | | |
| | | 15:8 | DCOEN | DCOFILTER[2:0] | | | LBYPASS | LTIME[2:0] | | |
| | | 23:16 | DIV[7:0] | | | | | | | |
| | | 31:24 | | | | | | DIV[10:8] | | |
| 0x50 | DPLL1SYNCBUSY | 7:0 | | | | | | DPLLRATIO | ENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x54 | DPLL1STATUS | 7:0 | | | | | | | CLKRDY | LOCK |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 28.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Write-protection is denoted by the "PAC Write-Protection" property in each individual register description. Refer to the Register Access Protection section and the PAC - Peripheral Access Controller chapter for details.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" or "Write-Synchronized" property in each individual register description. Refer to the section on Synchronization for details.

**Related Links**

Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.1. Event Control

**Name:** EVCTRL
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CFDEO1 | CFDEO0 |
| Access | R | R | R | R | R | R | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1 – CFDEO** Clock n Failure Detector Event Output Enable [n=0,1]
This bit indicates whether the XOSC Clock Failure detector event output is enabled whether and an output event will be generated when the XOSC Clock Failure detector detects a clock failure.
0: Clock Failure detector event output is disabled and an event will not be generated.
1: Clock Failure detector event output is enabled and an event will be generated.
To prevent false event generation, the bit CFDEOn must be set or cleared only when the XOSCn is disabled (XOSCCTRLn.ENABLE=0).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.2. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL1LDRTO | DPLL1LTO | DPLL1LCKF | DPLL1LCKR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL0LDRTO | DPLL0LTO | DPLL0LCKF | DPLL0LCKR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 27 – DPLL1LDRTO** DPLL1 Loop Divider Ratio Update Complete Interrupt Enable
0: The DPLL1 Loop Divider Ratio Update Complete interrupt is disabled.
1: The DPLL1 Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL1 Loop Divider Ratio Update Complete Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL1 Loop Divider Ratio Update Complete Interrupt Enable bit, which disables the DPLL1 Loop Divider Ratio Update Complete interrupt.

**Bit 26 – DPLL1LTO** DPLL1 Lock Timeout Interrupt Enable
0: The DPLL1 Lock Timeout interrupt is disabled.
1: The DPLL1 Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL1 Lock Timeout Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL1 Lock Timeout Interrupt Enable bit, which disables the DPLL1 Lock Timeout interrupt.

**Bit 25 – DPLL1LCKF** DPLL1 Lock Fall Interrupt Enable
0: The DPLL1 Lock Fall interrupt is disabled.
1: The DPLL1 Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL1 Lock Fall Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL1 Lock Fall Interrupt Enable bit, which disables the DPLL1 Lock Fall interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 24 – DPLL1LCKR** DPLL1 Lock Rise Interrupt Enable
0: The DPLL1 Lock Rise interrupt is disabled.
1: The DPLL1 Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL1 Lock Rise Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL1 Lock Rise Interrupt Enable bit, which disables the DPLL1 Lock Rise interrupt.

**Bit 19 – DPLL0LDRTO** DPLL0 Loop Divider Ratio Update Complete Interrupt Enable
0: The DPLL0 Loop Divider Ratio Update Complete interrupt is disabled.
1: The DPLL0 Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL0 Loop Divider Ratio Update Complete Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL0 Loop Divider Ratio Update Complete Interrupt Enable bit, which disables the DPLL0 Loop Divider Ratio Update Complete interrupt.

**Bit 18 – DPLL0LTO** DPLL0 Lock Timeout Interrupt Enable
0: The DPLL0 Lock Timeout interrupt is disabled.
1: The DPLL0 Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL0 Lock Timeout Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL0 Lock Timeout Interrupt Enable bit, which disables the DPLL0 Lock Timeout interrupt.

**Bit 17 – DPLL0LCKF** DPLL0 Lock Fall Interrupt Enable
0: The DPLL0 Lock Fall interrupt is disabled.
1: The DPLL0 Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL0 Lock Fall Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL0 Lock Fall Interrupt Enable bit, which disables the DPLL0 Lock Fall interrupt.

**Bit 16 – DPLL0LCKR** DPLL0 Lock Rise Interrupt Enable
0: The DPLL0 Lock Rise interrupt is disabled.
1: The DPLL0 Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL0 Lock Rise Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DPLL0 Lock Rise Interrupt Enable bit, which disables the DPLL0 Lock Rise interrupt.

**Bit 12 – DFLLRCS** DFLL Reference Clock Stopped Interrupt Enable
0: The DFLL Reference Clock Stopped interrupt is disabled.
1: The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

**Bit 11 – DFLLLCKC** DFLL Lock Coarse Interrupt Enable
0: The DFLL Lock Coarse interrupt is disabled.
1: The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Lock Coarse Interrupt Enable bit, which disables the DFLL Lock Coarse interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 10 – DFLLLCKF** DFLL Lock Fine Interrupt Enable
0: The DFLL Lock Fine interrupt is disabled.
1: The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Lock Fine Interrupt Enable bit, which disables the DFLL Lock Fine interrupt.

**Bit 9 – DFLLOOB** DFLL Out Of Bounds Interrupt Enable
0: The DFLL Out Of Bounds interrupt is disabled.
1: The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Out Of Bounds Interrupt Enable bit, which disables the DFLL Out Of Bounds interrupt.

**Bit 8 – DFLLRDY** DFLL Ready Interrupt Enable
0: The DFLL Ready interrupt is disabled.
1: The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Ready Interrupt Enable bit, which disables the DFLL Ready interrupt.

**Bits 2, 3 – XOSCFAIL** XOSC n Clock Failure Interrupt Enable
0: The XOSC n Clock Failure interrupt is disabled.
1: The XOSC0 Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC0 Clock Failure Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the XOSC n Clock Failure Interrupt Enable bit, which disables the XOSC n Clock Failure interrupt.

**Bits 0, 1 – XOSCRDY** XOSC n Ready Interrupt Enable
0: The XOSC n Ready interrupt is disabled.
1: The XOSC0 Ready interrupt is enabled, and an interrupt request will be generated when the XOSC n Ready Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the XOSC n Ready Interrupt Enable bit, which disables the XOSC n Ready interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.3. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL1LDRTO | DPLL1LTO | DPLL1LCKF | DPLL1LCKR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL0LDRTO | DPLL0LTO | DPLL0LCKF | DPLL0LCKR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 27 – DPLL1LDRTO** DPLL1 Loop Divider Ratio Update Complete Interrupt Enable
0: The DPLL1 Loop Divider Ratio Update Complete interrupt is disabled.
1: The DPLL1 Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL1 Loop Divider Ratio Update Complete Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL1 Loop Divider Ratio Update Complete Interrupt Enable bit, which enables the DPLL1 Loop Divider Ratio Update Complete interrupt.

**Bit 26 – DPLL1LTO** DPLL1 Lock Timeout Interrupt Enable
0: The DPLL1 Lock Timeout interrupt is disabled.
1: The DPLL1 Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL1 Lock Timeout Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL1 Lock Timeout Interrupt Enable bit, which enables the DPLL1 Lock Timeout interrupt.

**Bit 25 – DPLL1LCKF** DPLL1 Lock Fall Interrupt Enable
0: The DPLL1 Lock Fall interrupt is disabled.
1: The DPLL1 Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL1 Lock Fall Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL1 Lock Fall Interrupt Enable bit, which enables the DPLL1 Lock Fall interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 24 – DPLL1LCKR** DPLL1 Lock Rise Interrupt Enable
0: The DPLL1 Lock Rise interrupt is disabled.
1: The DPLL1 Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL1 Lock Rise Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL1 Lock Rise Interrupt Enable bit, which enables the DPLL1 Lock Rise interrupt.

**Bit 19 – DPLL0LDRTO** DPLL0 Loop Divider Ratio Update Complete Interrupt Enable
0: The DPLL0 Loop Divider Ratio Update Complete interrupt is disabled.
1: The DPLL0 Loop Divider Ratio Update Complete interrupt is enabled, and an interrupt request will be generated when the DPLL0 Loop Divider Ratio Update Complete Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL0 Loop Divider Ratio Update Complete Interrupt Enable bit, which enables the DPLL0 Loop Divider Ratio Update Complete interrupt.

**Bit 18 – DPLL0LTO** DPLL0 Lock Timeout Interrupt Enable
0: The DPLL0 Lock Timeout interrupt is disabled.
1: The DPLL0 Lock Timeout interrupt is enabled, and an interrupt request will be generated when the DPLL0 Lock Timeout Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL0 Lock Timeout Interrupt Enable bit, which enables the DPLL0 Lock Timeout interrupt.

**Bit 17 – DPLL0LCKF** DPLL0 Lock Fall Interrupt Enable
0: The DPLL0 Lock Fall interrupt is disabled.
1: The DPLL0 Lock Fall interrupt is enabled, and an interrupt request will be generated when the DPLL0 Lock Fall Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL0 Lock Fall Interrupt Enable bit, which enables the DPLL0 Lock Fall interrupt.

**Bit 16 – DPLL0LCKR** DPLL0 Lock Rise Interrupt Enable
0: The DPLL0 Lock Rise interrupt is disabled.
1: The DPLL0 Lock Rise interrupt is enabled, and an interrupt request will be generated when the DPLL0 Lock Rise Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DPLL0 Lock Rise Interrupt Enable bit, which enables the DPLL0 Lock Rise interrupt.

**Bit 12 – DFLLRCS** DFLL Reference Clock Stopped Interrupt Enable
0: The DFLL Reference Clock Stopped interrupt is disabled.
1: The DFLL Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DFLL Reference Clock Stopped Interrupt Enable bit, which enables the DFLL Reference Clock Stopped interrupt.

**Bit 11 – DFLLLCKC** DFLL Lock Coarse Interrupt Enable
0: The DFLL Lock Coarse interrupt is disabled.
1: The DFLL Lock Coarse interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Coarse Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DFLL Lock Coarse Interrupt Enable bit, which enables the DFLL Lock Coarse interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 10 – DFLLLCKF** DFLL Lock Fine Interrupt Enable
0: The DFLL Lock Fine interrupt is disabled.
1: The DFLL Lock Fine interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Fine Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DFLL Lock Fine Interrupt Disable/Enable bit, disable the DFLL Lock Fine interrupt and set the corresponding interrupt request.

**Bit 9 – DFLLOOB** DFLL Out Of Bounds Interrupt Enable
0: The DFLL Out Of Bounds interrupt is disabled.
1: The DFLL Out Of Bounds interrupt is enabled, and an interrupt request will be generated when the DFLL Out Of Bounds Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DFLL Out Of Bounds Interrupt Enable bit, which enables the DFLL Out Of Bounds interrupt.

**Bit 8 – DFLLRDY** DFLL Ready Interrupt Enable
0: The DFLL Ready interrupt is disabled.
1: The DFLL Ready interrupt is enabled, and an interrupt request will be generated when the DFLL Ready Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the DFLL Ready Interrupt Enable bit, which enables the DFLL Ready interrupt.

**Bits 2, 3 – XOSCFAIL** XOSCn Clock Failure Interrupt Enable
0: The XOSCn Clock Failure interrupt is disabled.
1: The XOSCn Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSCn Clock Failure Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the XOSCn Clock Failure Interrupt Enable bit, which enables the XOSCn Clock Failure interrupt.

**Bits 0, 1 – XOSCRDY** XOSCn Ready Interrupt Enable
0: The XOSCn Ready interrupt is disabled.
1: The XOSCn Ready interrupt is enabled, and an interrupt request will be generated when the XOSC0 Ready Interrupt flag is set.
Writing a zero to this bit has no effect.
Writing a '1' to this bit will set the XOSCn Ready Interrupt Enable bit, which enables the XOSCn Ready interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.4. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL1LDRTO | DPLL1LTO | DPLL1LCKF | DPLL1LCKR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL0LDRTO | DPLL0LTO | DPLL0LCKF | DPLL0LCKR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 27 – DPLL1LDRTO** DPLL1 Loop Divider Ratio Update Complete
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL1 Loop Divider Ratio Update Complete bit in the Status register (STATUS.DPLL1LDRTO) and will generate an interrupt request if INTENSET.DPLL1LDRTO is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DPLL1 Loop Divider Ratio Update Complete interrupt flag.

**Bit 26 – DPLL1LTO** DPLL1 Lock Timeout
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL1 Lock Timeout bit in the Status register (STATUS. DPLL1LTO) and will generate an interrupt request if INTENSET.DPLL1LTO is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DPLL1 Lock Timeout interrupt flag.

**Bit 25 – DPLL1LCKF** DPLL1 Lock Fall
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL1 Lock Fall bit in the Status register (STATUS.DPLL1LCKF) and will generate an interrupt request if INTENSET.DPLL1LCKF is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DPLL1 Lock Fall interrupt flag.

**Bit 24 – DPLL1LCKR** DPLL1 Lock Rise
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL1 Lock Rise bit in the Status register (STATUS. DPLL1LCKR) and will generate an interrupt request if INTENSET.DPLL1LCKR is '1'.
Writing a zero to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit clears the DPLL1 Lock Rise interrupt flag.

**Bit 19 – DPLL0LDRTO** DPLL0 Loop Divider Ratio Update Complete
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL0 Loop Divider Ratio Update Complete bit in the Status register (STATUS.DPLL0LDRTO) and will generate an interrupt request if INTENSET.DPLL0LDRTO is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DPLL0 Loop Divider Ratio Update Complete interrupt flag.

**Bit 18 – DPLL0LTO** DPLL0 Lock Timeout
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL0 Lock Timeout bit in the Status register (STATUS. DPLL0LTO) and will generate an interrupt request if INTENSET.DPLL0LTO is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DPLL0 Lock Timeout interrupt flag.

**Bit 17 – DPLL0LCKF** DPLL0 Lock Fall
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL0 Lock Fall bit in the Status register (STATUS.DPLL0LCKF) and will generate an interrupt request if INTENSET.DPLL0LCKF is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DPLL0 Lock Fall interrupt flag.

**Bit 16 – DPLL0LCKR** DPLL0 Lock Rise
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DPLL0 Lock Rise bit in the Status register (STATUS. DPLL0LCKR) and will generate an interrupt request if INTENSET.DPLL0LCKR is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DPLL0 Lock Rise interrupt flag.

**Bit 12 – DFLLRCS** DFLL Reference Clock Stopped
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Reference Clock Stopped bit in the Status register (STATUS. DFLLRCS) and will generate an interrupt request if INTENSET.DFLLRCS is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DFLL Reference Clock Stopped interrupt flag.

**Bit 11 – DFLLLCKC** DFLL Lock Coarse
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Lock Coarse bit in the Status register (STATUS.DFLLLCKC) and will generate an interrupt request if INTENSET.DFLLLCKC is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DFLL Lock Coarse interrupt flag.

**Bit 10 – DFLLLCKF** DFLL Lock Fine
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Lock Fine bit in the Status register (STATUS.DFLLLCKF) and will generate an interrupt request if INTENSET.DFLLLCKF is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DFLL Lock Fine interrupt flag.

**Bit 9 – DFLLOOB** DFLL Out Of Bounds
This flag is cleared by writing a '1' to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This flag is set on a zero-to-one transition of the DFLL Out Of Bounds bit in the Status register (STATUS.DFLLOOB) and will generate an interrupt request if INTENSET.DFLLOOB is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DFLL Out Of Bounds interrupt flag.

**Bit 8 – DFLLRDY** DFLL Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Ready bit in the Status register (STATUS.DFLLRDY) and will generate an interrupt request if INTENSET.DFLLRDY is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the DFLL Ready interrupt flag.

**Bits 2, 3 – XOSCFAIL** XOSCn Clock Failure
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSCn Clock Failure bit in the Status register (STATUS.XOSCFAILn) and will generate an interrupt request if INTENSET.XOSCFAILn is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the XOSCn Clock Failure interrupt flag.

**Bits 0, 1 – XOSCRDY** XOSCn Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC0 Ready bit in the Status register (STATUS.XOSCRDYn) and will generate an interrupt request if INTENSET.XOSCRDYn is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the XOSCn Ready interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.5. Status

**Name:** STATUS
**Offset:** 0x10
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL1LDRTO | DPLL1TO | DPLL1LCKF | DPLL1LCKR |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DPLL0LDRTO | DPLL0TO | DPLL0LCKF | DPLL0LCKR |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | DFLLRCS | DFLLLCKC | DFLLLCKF | DFLLOOB | DFLLRDY |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | XOSCCKSW1 | XOSCCKSW0 | XOSCFAIL1 | XOSCFAIL0 | XOSCRDY1 | XOSCRDY0 |
| Access | | | R | R | R | R | R | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 27 – DPLL1LDRTO** DPLL1 Loop Divider Ratio Update Complete
0: DPLL1 Loop Divider Ratio Update Complete not detected.
1: DPLL1 Loop Divider Ratio Update Complete detected.

**Bit 26 – DPLL1TO** DPLL1 Lock Timeout
0: DPLL1 Lock time-out not detected.
1: DPLL1 Lock time-out detected.

**Bit 25 – DPLL1LCKF** DPLL1 Lock Fall
0: DPLL1 Lock fall edge not detected.
1: DPLL1 Lock fall edge detected.

**Bit 24 – DPLL1LCKR** DPLL1 Lock Rise
0: DPLL1 Lock rise edge not detected.
1: DPLL1 Lock rise edge detected.

**Bit 19 – DPLL0LDRTO** DPLL0 Loop Divider Ratio Update Complete
0: DPLL0 Loop Divider Ratio Update Complete not detected.
1: DPLL0 Loop Divider Ratio Update Complete detected.

**Bit 18 – DPLL0TO** DPLL0 Lock Timeout
0: DPLL0 Lock time-out not detected.
1: DPLL0 Lock time-out detected.

**Bit 17 – DPLL0LCKF** DPLL0 Lock Fall
0: DPLL0 Lock fall edge not detected.
1: DPLL0 Lock fall edge detected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 16 – DPLL0LCKR** DPLL0 Lock Rise
0: DPLL0 Lock rise edge not detected.
1: DPLL0 Lock rise edge detected.

**Bit 12 – DFLLRCS** DFLL Reference Clock Stopped
0: DFLL reference clock is running.
1: DFLL reference clock has stopped.

**Bit 11 – DFLLLCKC** DFLL Lock Coarse
0: No DFLL coarse lock detected.
1: DFLL coarse lock detected.

**Bit 10 – DFLLLCKF** DFLL Lock Fine
0: No DFLL fine lock detected.
1: DFLL fine lock detected.

**Bit 9 – DFLLOOB** DFLL Out Of Bounds
0: No DFLL Out Of Bounds detected.
1: DFLL Out Of Bounds detected.

**Bit 8 – DFLLRDY** DFLL Ready
0: DFLL is not ready.
1: DFLL is stable and ready to be used as a clock source.

**Bit 5 – XOSCCKSW1** XOSC1 Clock Switch
0: XOSC1 is not switched and provides the external clock or crystal oscillator clock.
1: XOSC is switched and provides the safe clock.

**Bit 4 – XOSCCKSW0** XOSC0 Clock Switch
0: XOSC0 is not switched and provides the external clock or crystal oscillator clock.
1: XOSC0 is switched and provides the safe clock.

**Bit 3 – XOSCFAIL1** XOSC1 Clock Failure
0: XOSC1 failure not detected.
1: XOSC1 failure detected.

**Bit 2 – XOSCFAIL0** XOSC0 Clock Failure
0: XOSC0 failure not detected.
1: XOSC0 failure detected.

**Bit 1 – XOSCRDY1** XOSC1 Ready
0: XOSC1 is not ready.
1: XOSC1 is stable and ready to be used as a clock source.

**Bit 0 – XOSCRDY0** XOSC0 Ready
0: XOSC0 is not ready.
1: XOSC0 is stable and ready to be used as a clock source.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.6. External Multipurpose Crystal Oscillator Control

**Name:** XOSCCTRL
**Offset:** 0x14 + n*0x04 [n=0..1]
**Reset:** 0x00000080
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CFDPRESC[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | STARTUP[3:0] | | | | | | SWBEN | CFDEN |
| Access | R/W | R/W | R/W | R/W | | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ENALC | IMULT[3:0] | | | | IPTAT[1:0] | | LOWBUFGAIN |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | | | XTALEN | ENABLE | |
| Access | R/W | R/W | | | | R/W | R/W | |
| Reset | 1 | 0 | | | | 0 | 0 | |

**Bits 27:24 – CFDPRESC[3:0]** Clock Failure Detector Prescaler
These bits select the prescaler for the clock failure detector.
The DFLL48 oscillator is used to clock the CFD prescaler.
The CFD safe clock frequency is the DFLL48 frequency divided by 2^CFDPRESC.

**Bits 23:20 – STARTUP[3:0]** Start-Up Time
These bits select start-up time for the oscillator XOSCn according to the table below.
The OSCULP32K oscillator is used to clock the start-up counter.

**Table 28-6.** Start-UpTime for External Multipurpose Crystal Oscillator

| STARTUP[3:0] | Number of OSCULP32K Clock Cycles | Number of XOSC Clock Cycles | Approximate Equivalent Time( |
|---|---|---|---|
| 0x0 | 1 | 3 | 31µs |
| 0x1 | 2 | 3 | 61µs |
| 0x2 | 4 | 3 | 122µs |
| 0x3 | 8 | 3 | 244µs |
| 0x4 | 16 | 3 | 488µs |
| 0x5 | 32 | 3 | 977µs |
| 0x6 | 64 | 3 | 1953µs |
| 0x7 | 128 | 3 | 3906µs |
| 0x8 | 256 | 3 | 7813µs |
| 0x9 | 512 | 3 | 15625µs |
| 0xA | 1024 | 3 | 31250µs |
| 0xB | 2048 | 3 | 62500µs |
| 0xC | 4096 | 3 | 125000µs |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 28-6.** Start-UpTime for External Multipurpose Crystal Oscillator (continued)

| STARTUP[3:0] | Number of OSCULP32K Clock Cycles | Number of XOSC Clock Cycles | Approximate Equivalent Time( |
|---|---|---|---|
| 0xD | 8192 | 3 | 250000µs |
| 0xE | 16384 | 3 | 500000µs |
| 0xF | 32768 | 3 | 1000000µs |

**Bit 17 – SWBEN** Xosc Clock Switch Enable
This bit controls the XOSCn output clock switch back to the external clock or crystal oscillator in case of clock recovery :
0: The clock switch back is disabled.
1: The clock switch back is enabled. This bit is reset once the XOSCn output clock is switched back to the external clock or crystal oscillator.

**Bit 16 – CFDEN** Clock Failure Detector Enable
This bit controls the XOSCn clock failure detector :
0: the Clock Failure Detector is disabled.
1: the Clock Failure Detector is enabled.

**Bit 15 – ENALC** Automatic Loop Control Enable
This bit controls the XOSCn automatic loop control :
0: the automatic loop control is disabled.
1: the automatic loop control is enabled. Oscillator's amplitude will be automatically adjusted during Crystal Oscillator operation.

**Bits 14:11 – IMULT[3:0]** Oscillator Current Multiplier
These bits select the current multiplier for the oscillator XOSCn, given in table External Multipurpose Crystal Oscillator Current Settings.

**Bits 10:9 – IPTAT[1:0]** Oscillator Current Reference
These bits select the current reference for the oscillator XOSCn, given in table below.

**Table 28-7.** External Multipurpose Crystal Oscillator Current Settings

| Frequency Range | Current Setting | |
|---|---|---|
| | IMULT[3:0] | IPTAT[1:0] |
| >24MHz to 48MHz | 6 | 3 |
| >16MHz to 24MHz | 5 | 3 |
| >8MHz to 16MHz | 4 | 3 |
| 8MHz | 3 | 2 |

For relatively small CLOAD in a frequency range, the setting for the lower frequency range can be used to preserve current consumption.

**Bit 8 – LOWBUFGAIN** Low Buffer Gain Enable
0: The low buffer gain of oscillator XOSCn is disabled.
1: The low buffer gain of oscillator XOSCn is enabled.
When XOSCCTRLn.ENALC=0 this bit has no effect.
When XOSCCTRLn.ENALC=1, this bit is used to adjust the oscillator's amplitude in automatic loop control.
The default value of LOWBUFGAIN=0 should be used to allow operating with a low amplitude oscillator. Use this setting except to solve stability issues.
Setting LOWBUFGAIN=1 will increase the oscillator's amplitude by a factor of approximately 2. Use this setting to solve stability issues.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 7 – ONDEMAND** On Demand Control

The On Demand operation mode allows the oscillator XOSCn to be enabled or disabled, depending on peripheral clock requests.
If On Demand is set, the oscillator will be running only when requested by a peripheral and enabled (XOSCCTRLn. ENABLE=1). If there is no peripheral requesting the oscillator's clock source, the oscillator will be in a disabled state.
If On Demand is cleared, the oscillator will always be running when enabled (XOSCCTRLn.ENABLE=1). In standby sleep mode, the On Demand operation is still active.
0: The oscillator is always on.
1: The oscillator is running when a peripheral is requesting the oscillator to be used as a clock source. The oscillator is not running if no peripheral is requesting the clock source.

**Bit 6 – RUNSTDBY** Run in Standby

This bit controls how the XOSCn behaves during standby sleep mode:
0: The XOSCn is not running in standby sleep mode if no peripheral requests the clock.
1: The XOSCn is running in standby sleep mode. If ONDEMAND is one, the XOSCn will be running when a peripheral is requesting the clock. If ONDEMAND is zero, the clock source will always be running in standby sleep mode.

**Bit 2 – XTALEN** Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator XOSCn:
0: External clock connected on XIN. XOUT can be used as general-purpose I/O.
1: Crystal connected to XIN/XOUT.

**Bit 1 – ENABLE** Oscillator Enable

0: The oscillator XOSCn is disabled.
1: The oscillator XOSCn is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.7. DFLL48M Control A

**Name:** DFLLCTRLA
**Offset:** 0x1C
**Reset:** 0x82
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | | | | ENABLE | |
| Access | R/W | R/W | | | | | R/W | |
| Reset | 1 | 0 | | | | | 1 | |

**Bit 7 – ONDEMAND** On Demand Control
The On Demand operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests.
If On Demand is set, the DFLL will only be running when requested by a peripheral and enabled (DFLLTRLA. ENABLE=1). If there is no peripheral requesting the DFLL's clock source, the DFLL will be in a disabled state.
If On Demand is disabled the DFLL will always be running when enabled (DFLLTRLA.ENABLE=1). In standby sleep mode, the On Demand operation is still active.
0: The DFLL is always on.
1: The DFLL is running when a peripheral is requesting the DFLL to be used as a clock source. The DFLL is not running if no peripheral is requesting the clock source.

**Bit 6 – RUNSTDBY** Run in Standby
This bit controls how the DFLL behaves during standby sleep mode:
0: The DFLL is not running in standby sleep mode if no peripheral requests the clock.
1: The DFLL is running in standby sleep mode. If ONDEMAND is one, the DFLL will be running when a peripheral is requesting the clock. If ONDEMAND is zero, the clock source will always be running in standby sleep mode.

**Bit 1 – ENABLE** DFLL Enable
0: The DFLL oscillator is disabled.
1: The DFLL oscillator is enabled.
**Note:** This bit is write-synchronized: Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to DFLLCTRLA.ENABLE will read back immediately after written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.8. DFLL48M Control B

**Name:** DFLLCTRLB
**Offset:** 0x20
**Reset:** 0x00
**Property:** Read-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WAITLOCK | BPLCKC | QLDIS | CCDIS | USBCRM | LLAW | STABLE | MODE |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – WAITLOCK** Wait Lock
This bit controls the DFLL output clock, depending on lock status:
0: Output clock before the DFLL is locked.
1: Output clock when DFLL is locked (Fine lock).

**Bit 6 – BPLCKC** Bypass Coarse Lock
This bit controls the coarse lock procedure:
0: Bypass coarse lock is disabled.
1: Bypass coarse lock is enabled.

**Bit 5 – QLDIS** Quick Lock Disable
0: Quick Lock is enabled.
1: Quick Lock is disabled.

**Bit 4 – CCDIS** Chill Cycle Disable
0: Chill Cycle is enabled.
1: Chill Cycle is disabled.

**Bit 3 – USBCRM** USB Clock Recovery Mode
0: USB Clock Recovery Mode is disabled.
1: USB Clock Recovery Mode is enabled.

**Bit 2 – LLAW** Lose Lock After Wake
0: Locks will not be lost after waking up from sleep modes if the DFLL clock has been stopped.
1: Locks will be lost after waking up from sleep modes if the DFLL clock has been stopped.

**Bit 1 – STABLE** Stable DFLL Frequency
0: FINE calibration tracks changes in output frequency.
1: FINE calibration register value will be fixed after a fine lock.

**Bit 0 – MODE** Operating Mode Selection
0: The DFLL operates in open-loop operation.
1: The DFLL operates in closed-loop operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.9. DFLL48M Value

**Name:** DFLLVAL
**Offset:** 0x24
**Reset:** 0x0000XXXX
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIFF[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DIFF[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COARSE[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | x | x | x | x | 0 | 0 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FINE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | 0 | 0 | 0 | 0 |

**Bits 31:16 – DIFF[15:0]** Multiplication Ratio Difference
In closed-loop mode (DFLLCTRLB.MODE is written to one), this bit group indicates the difference between the ideal number of DFLL cycles and the counted number of cycles. This value is not updated in open-loop mode, and should be considered invalid in that case.

**Bits 15:10 – COARSE[5:0]** Coarse Value
Set the value of the Coarse Calibration register. In closed-loop mode, this field is read-only.
The DFLL48M is factory-calibrated for 48MHz. Register DFLLVAL.COARSE stores the coarse frequency calibration after reset.

**Bits 7:0 – FINE[7:0]** Fine Value
Set the value of the Fine Calibration register. In closed-loop mode, this field is read-only.
The DFLL48M is factory-calibrated for 48MHz. Register DFLLVAL.FINE stores the coarse frequency calibration after reset.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.10. DFLL48M Multiplier

**Name:** DFLLMUL
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CSTEP[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FSTEP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MUL[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MUL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:26 – CSTEP[5:0]** Coarse Maximum Step
This bit group indicates the maximum step size allowed during coarse adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

**Bits 23:16 – FSTEP[7:0]** Fine Maximum Step
This bit group indicates the maximum step size allowed during fine adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

**Bits 15:0 – MUL[15:0]** DFLL Multiply Factor
This field determines the ratio of the CLK_DFLL output frequency to the CLK_DFLL_REF input frequency. Writing to the MUL bits will cause locks to be lost and the fine calibration value to be reset to its midpoint.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.11. DFLL48M Synchronization

**Name:** DFLLSYNC
**Offset:** 0x2C
**Reset:** 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | DFLLMUL | DFLLVAL | DFLLCTRLB | ENABLE | |
| Access | | | | R | R | R | R | |
| Reset | | | | 0 | 0 | 0 | 0 | |

**Bit 4 – DFLLMUL** DFLLMUL Synchronization Busy
This bit is cleared when the synchronization of DFLLMUL register between the clock domains is complete.
This bit is set when the synchronization of DFLLMUL register between clock domains is started.
The DFLLMUL synchronization only applies for write operations.

**Bit 3 – DFLLVAL** DFLLVAL Synchronization Busy
This bit is cleared when the synchronization of DFLLVAL register between the clock domains is complete.
This bit is set when the synchronization of DFLLVAL register between clock domains is started.
The DFLLVAL synchronization applies for read and write operations.

**Bit 2 – DFLLCTRLB** DFLLCTRLB Synchronization Busy
This bit is cleared when the synchronization of DFLLCTRLB register between the clock domains is complete.
This bit is set when the synchronization of DFLLCTRLB register between clock domains is started.
The DFLLCTRLB synchronization only applies for write operations.

**Bit 1 – ENABLE** ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE register bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE register bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.12. DPLL Control A

**Name:** DPLLCTRLA
**Offset:** 0x30 + n*0x14 [n=0..1]
**Reset:** 0x80
**Property:** PAC Write-Protection, Write-Synchronized(ENABLE), Enable-Protected (ONDEMAND, RUNSTDBY)

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | | | | ENABLE | |
| Access | R/W | R/W | | | | | R/W | |
| Reset | 1 | 0 | | | | | 0 | |

**Bit 7 – ONDEMAND** On Demand Control
The On Demand operation mode allows the DPLLn to be enabled or disabled, depending on peripheral clock requests.
If On Demand is set, the DPLLn will be running only when requested by a peripheral and enabled (DPLLnCTRLA. ENABLE=1). If there is no peripheral requesting the DPLLn's clock source, the DPLLn will be in a disabled state.
If On Demand is cleared, the DPLLn will always be running when enabled (DPLLnCTRLA.ENABLE=1). In standby sleep mode, the On Demand operation is still active.
0: The DPLLn is always running.
1: The DPLLn is running when a peripheral is requesting the DPLLn to be used as a clock source. The DPLLn is not running if no peripheral is requesting the clock source.

**Bit 6 – RUNSTDBY** Run in Standby
This bit controls how the DPLLn behaves during standby sleep mode:
0: The DPLLn is not running in standby sleep mode if no peripheral requests the clock.
1: The DPLLn is running in standby sleep mode. If ONDEMAND is one, the DPLLn will be running when a peripheral is requesting the clock. If ONDEMAND is zero, the clock source will always be running in standby sleep mode.

**Bit 1 – ENABLE** DPLL Enable
0: The DPLLn is disabled.
1: The DPLLn is enabled.
The software operation of enabling or disabling the DPLLn takes a few clock cycles, so the DPLLnSYNCBUSY. ENABLE status bit indicates when the DPLLn is successfully enabled or disabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.13. DPLL Ratio Control

**Name:** DPLLRATIO
**Offset:** 0x34 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

Refer to the Synchronization section in the Clock System Overview chapter for details on the functionality of this register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | LDRFRAC[4:0] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | LDR[12:8] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 20:16 – LDRFRAC[4:0]** Loop Divider Ratio Fractional Part
Write these bits to set the fractional part of the frequency multiplier. Due to synchronization there is a delay between writing to DPLLnRATIO.LDRFRAC[4:0] and the effect on the DPLLn output clock. The value written DPLLnRATIO.LDRFRAC[4:0] will be read back immediately and the DPLLRATIO bit in the synchronization busy register, DPLLnSYNCBUSY.DPLLRATIO, will be set. DPLLnSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.

**Bits 12:0 – LDR[12:0]** Loop Divider Ratio
Write these bits to set the integer part of the frequency multiplier. The value written DPLLnRATIO.LDR[3:0] will be read back immediately and the DPLLRATIO bit in the synchronization busy register, DPLLnSYNCBUSY.DPLLRATIO, will be set. DPLLnSYNCBUSY.DPLLRATIO will be cleared when the operation is completed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.14. DPLL Control B

**Name:** DPLLCTRLB
**Offset:** 0x38 + n*0x04 [n=0..1]
**Reset:** 0x00000020
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | DIV[10:8] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DIV[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DCOEN | DCOFILTER[2:0] | | | LBYPASS | LTIME[2:0] | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | REFCLK[2:0] | | | WUF | FILTER[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**Bits 26:16 – DIV[10:0]** Clock Divider
These bits are used to set the XOSC clock division factor and can be calculated with following formula:

$$f_{DIV} = \frac{f\text{XOSC}}{2 \times (\text{DIV} + 1)}$$

**Bit 15 – DCOEN** DCO Filter Enable
0: Disable DCO filter controller. Sigma-Delta DAC is automatically set the PLL itself.
1: Enable DCO filter controller. DCOFILTER[2:0] is used to select sigma-delta DAC filter bandwidth.

**Bits 14:12 – DCOFILTER[2:0]** Sigma-Delta DCO Filter Selection
These bits select the DPLLn sigma-delta DCO filter type, as shown in the table below:

**Table 28-8.** Sigma-delta DCO Filter selection

| DCOFILTER[2:0] | Capacitor (pF) | Bandwidth Fn (MHz) |
|---|---|---|
| 0x0 | 0.5 | 3.21 |
| 0x1 | 1 | 1.6 |
| 0x2 | 1.5 | 1.1 |
| 0x3 | 2 | 0.8 |
| 0x4 | 2.5 | 0.64 |
| 0x5 | 3 | 0.55 |
| 0x6 | 3.5 | 0.45 |
| 0x7 | 4 | 0.4 |

**Bit 11 – LBYPASS** Lock Bypass


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 10:8 – LTIME[2:0]** Lock Time

Write these bits to select the lock time-out value, as shown in the figure below:

| Value | Name | Description |
|---|---|---|
| 0x0 | Default | No time-out. Automatic lock. |
| 0x1 | Reserved | |
| 0x2 | Reserved | |
| 0x3 | Reserved | |
| 0x4 | 800US | Time-out if no lock within 800 us |
| 0x5 | 900US | Time-out if no lock within 900 us |
| 0x6 | 1MS | Time-out if no lock within 1 ms |
| 0x7 | 1P1MS | Time-out if no lock within 1.1 ms |

**Bits 7:5 – REFCLK[2:0]** Reference Clock Selection

Write these bits to select the DPLLn clock reference, as shown in the table below:

| Value | Name | Description |
|---|---|---|
| 0x0 | GCLK | Dedicated GCLK clock reference |
| 0x1 | XOSC32 | XOSC32K clock reference (default) |
| 0x2 | XOSC0 | XOSC0 clock reference |
| 0x3 | XOSC1 | XOSC1 clock reference |
| Other | - | Reserved |

**Bit 4 – WUF** Wake Up Fast

0: DPLLn clock is output after startup and lock time.
1: DPLLn clock is output after startup time.

**Bits 3:0 – FILTER[3:0]** Proportional Integral Filter Selection

These bits select the DPLLn digital filter type, as shown in the table below:

**Table 28-9.** Proportional Integral Filter selection

| FILTER[3:0] | PLL Bandwidth (fn) | Damping Factor |
|---|---|---|
| 0x0 | 92.7 kHz | 0.76 |
| 0x1 | 131 kHz | 1.08 |
| 0x2 | 46.4 kHz | 0.38 |
| 0x3 | 65.6 kHz | 0.54 |
| 0x4 | 131 kHz | 0.56 |
| 0x5 | 185 kHz | 0.79 |
| 0x6 | 65.6 kHz | 0.28 |
| 0x7 | 92.7 kHz | 0.39 |
| 0x8 | 46.4 kHz | 1.49 |
| 0x9 | 65.6 kHz | 2.11 |
| 0xA | 23.2 kHz | 0.75 |
| 0xB | 32.8 kHz | 1.06 |
| 0xC | 65.6 kHz | 1.07 |
| 0xD | 92.7 kHz | 1.51 |
| 0xE | 32.8 kHz | 0.53 |
| 0xF | 46.4 kHz | 0.75 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.15. DPLL Synchronization Busy

**Name:** DPLLSYNCBUSY
**Offset:** 0x3C + n*0x04 [n=0..1]
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | DPLLRATIO | ENABLE | |
| Access | | | | | | R | R | |
| Reset | | | | | | 0 | 0 | |

**Bit 2 – DPLLRATIO** DPLL Loop Divider Ratio Synchronization Status
0: The DPLLRATIO register has been synchronized.
1: The DPLLRATIO register value has changed and its synchronization is in progress.

**Bit 1 – ENABLE** DPLL Enable Synchronization Status
0: The DPLLnCTRLA.ENABLE bit has been synchronized.
1: The DPLLnCTRLA.ENABLE bit value has changed and its synchronization is in progress.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 28.8.16. DPLL Status

**Name:** DPLLSTATUS
**Offset:** 0x40 + n*0x04 [n=0..1]
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CLKRDY | LOCK |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – CLKRDY** DPLL Clock Ready
0: The DPLLn output clock is off.
1: The DPLLn output clock in on.

**Bit 0 – LOCK** DPLL Lock Status
0: The DPLLn Lock signal is cleared, when the DPLLn is disabled or when the DPLLn is trying to reach the target frequency.
1: The DPLLn Lock signal is asserted when the desired frequency is reached.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 29. OSC32KCTRL – 32KHz Oscillators Controller

## 29.1. Overview

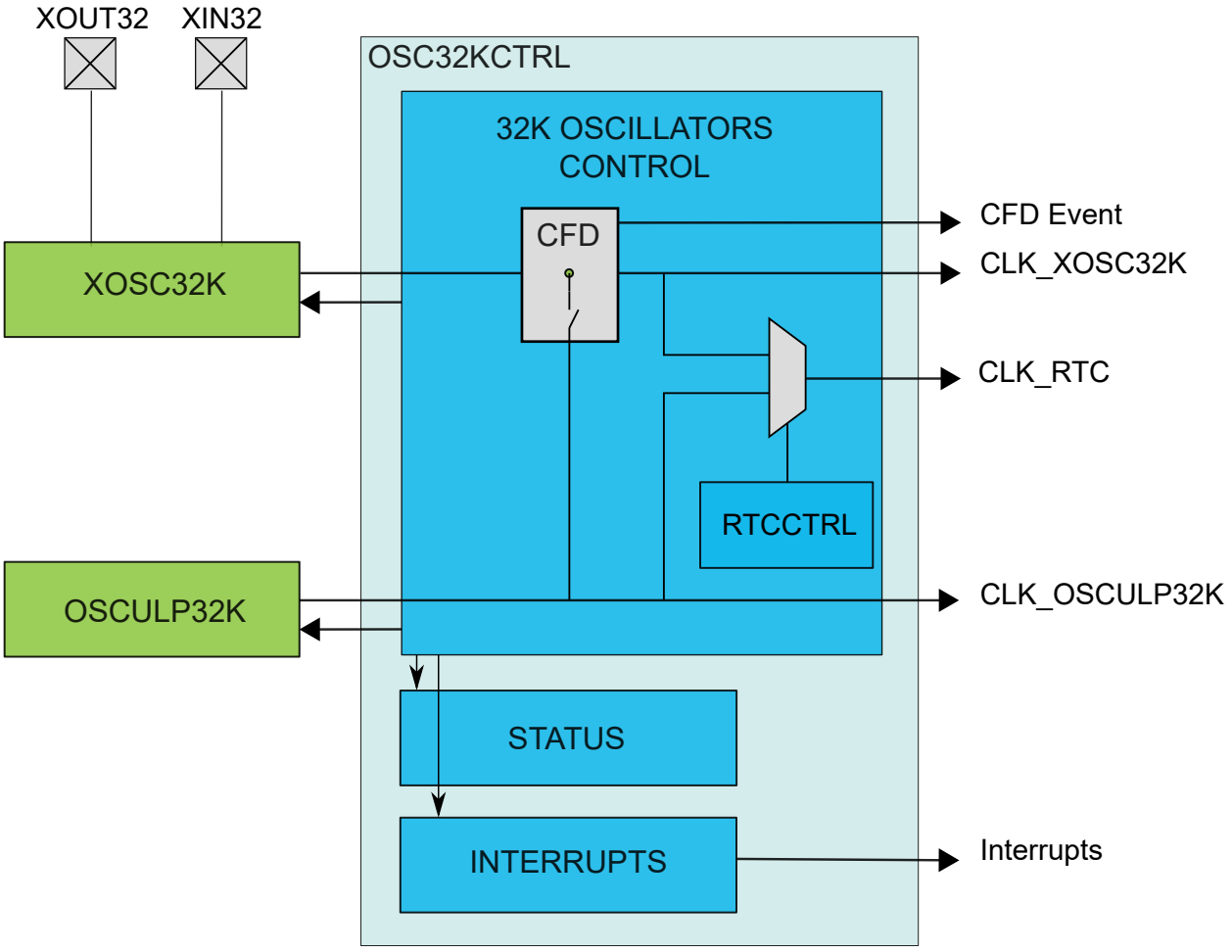
The 32.768 kHz Oscillators Controller (OSC32KCTRL) provides a user interface to the 32.768 kHz oscillators:

- A 32.768 kHz crystal oscillator (XOSC32K)
- A 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)

## 29.2. Features

- 32.768 kHz Crystal Oscillator (XOSC32K)
  - Programmable start-up time
  - Crystal or external input clock on XIN32 I/O
  - Clock failure detection with safe clock switch
  - Clock failure event output
- 32.768 kHz Ultra Low-Power Internal Oscillator (OSCULP32K)
  - Ultra low-power, always-on oscillator
- Calibration value loaded from Flash factory calibration at Reset
- 1.024 kHz clock outputs available

## 29.3. Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 29.4. Signal Description

| Signal | Description | Type |
|---|---|---|
| XIN32 | Analog Input | 32.768 kHz Crystal Oscillator or external clock input |
| XOUT32 | Analog Output | 32.768 kHz Crystal Oscillator output |

The I/O lines are automatically selected when XOSC32K is enabled.

**Note:** The signal of the external crystal oscillator may affect the jitter of neighboring pads.

## 29.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 29.5.1. I/O Lines

I/O lines are configured by OSC32KCTRL when XOSC32K is enabled and need no user configuration.

### 29.5.2. Power Management

The OSC32KCTRL will continue to operate in any sleep mode where a 32KHz oscillator is running as source clock. The OSC32KCTRL interrupts can be used to wake up the device from sleep modes.

**Related Links**

PM – Power Manager

### 29.5.3. Clocks

The OSC32KCTRL gathers controls for both 32 KHz oscillators and provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), and Watchdog Timer (WDT).

The available clock sources are: XOSC32K and OSCULP32K.

The OSC32KCTRL bus clock (CLK_OSC32KCTRL_APB) can be enabled and disabled in the Main Clock module (MCLK).

### 29.5.4. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the OSC32KCTRL interrupts requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 29.5.5. Events

The events of this peripheral are connected to the Event System.

**Related Links**

Event System (EVSYS)

### 29.5.6. Debug Operation

When the CPU is halted in debug mode, OSC32KCTRL will continue normal operation. If OSC32KCTRL is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

### 29.5.7. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 29.5.8. Analog Connections

The external 32.768 kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors. For details on recommended oscillator characteristics and capacitor load, refer to the related links.

## 29.6. Functional Description

### 29.6.1. Principle of Operation

XOSC32K and OSCULP32K are configured via OSC32KCTRL control registers.

The STATUS register gathers different status signals coming from the sub-peripherals of OSC32KCTRL. The status signals can be used to generate system interrupts, and in some cases wake up the system from standby mode, provided the corresponding interrupt is enabled.

### 29.6.2. 32 kHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in two different modes:

- External clock, with an external clock signal connected to XIN32
- Crystal oscillator, with an external 32.768 kHz crystal connected between XIN32 and XOUT32

At reset, the XOSC32K is disabled, and the XIN32/XOUT32 pins can either be used as General Purpose I/O (GPIO) pins or by other peripherals in the system.

When XOSC32K is enabled, the operating mode determines the GPIO usage. When in crystal oscillator mode, the XIN32 and XOUT32 pins are controlled by the OSC32KCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN32 pin will be overridden and controlled by the OSC32KCTRL, while the XOUT32 pin can still be used as a GPIO pin.

| | |
|---|---|
| **Enabling, Disabling** | The XOSC32K is enabled by writing a '1' to the Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 1).<br>The XOSC32K is disabled by writing a '0' to the Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 0). |
| **Mode Selection** | To enable the XOSC32K in Crystal Oscillator mode, the XTALEN bit in the 32 kHz External Crystal Oscillator Control register must be written (XOSC32K.XTALEN = 1). If XOSC32K.XTALEN is '0', the External Clock Input mode will be enabled. |
| **Gain Selection** | When a crystal oscillator is selected, a controllable gain is provided. Writing to the Control Gain Mode bit field (XOSC32K.CGM) will select a gain setting appropriate for the desired trade-off between low power and high speed. |
| **32KHz and 1KHz Output** | The XOSC32K 32.768 kHz output is enabled by setting the 32 kHz Output Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.EN32K=1). The XOSC32K also has a 1.024 kHz clock output. This is enabled by setting the 1 kHz Output Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.EN1K = 1). |
| **Configuration Lock** | It is also possible to lock the XOSC32K configuration by setting the Write Lock bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.WRTLOCK=1). If set, the XOSC32K configuration is locked until a Power-On Reset (POR) is detected. |

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.RUNSTDBY, XOSC32K.ONDEMAND, and XOSC32K.ENABLE. If XOSC32KCTRL.ENABLE = 0, the XOSC32K will be always stopped. For XOS32KCTRL.ENABLE = 1, this table is valid:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 29-1.** XOSC32K Sleep Behavior

| CPU Mode | XOSC32K.RUNSTDBY | XOSC32K.ONDEMAND | Sleep Behavior of XOSC32K and CFD |
|---|---|---|---|
| Active or Idle | - | 0 | Always run |
| Active or Idle | - | 1 | Run if requested by peripheral |
| Standby | 1 | 0 | Always run |
| Standby | 1 | 1 | Run if requested by peripheral |
| Standby | 0 | - | Run if requested by peripheral |

As a crystal oscillator usually requires a very long start-up time, the 32KHz External Crystal Oscillator will keep running across resets when XOSC32K.ONDEMAND=0, except for power-on reset (POR). After a reset or when waking up from a sleep mode where the XOSC32K was disabled, the XOSC32K will need time to stabilize on the correct frequency. This start-up time can be configured by changing the Oscillator Start-Up Time bit group (XOSC32K.STARTUP) in the 32 kHz External Crystal Oscillator Control register. During the start-up time, the oscillator output is masked to ensure that no unstable clock propagates to the digital logic.

Once the external clock or crystal oscillator is stable and ready to be used as a clock source, the XOSC32K Ready bit in the Status register is set (STATUS.XOSC32KRDY=1). The transition of STATUS.XOSC32KRDY from '0' to '1' generates an interrupt if the XOSC32K Ready bit in the Interrupt Enable Set register is set (INTENSET.XOSC32KRDY=1).

The XOSC32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). Before enabling the GCLK or the RTC module, the corresponding oscillator output must be enabled (XOSC32K.EN32K or XOSC32K.EN1K) in order to ensure proper operation. In the same way, the GCLK or RTC modules must be disabled before the clock selection is changed. For details on RTC clock configuration, refer also to Real-Time Counter Clock Selection.

**Related Links**

GCLK - Generic Clock Controller

RTC – Real-Time Counter

### 29.6.3. Clock Failure Detection Operation

The Clock Failure Detector (CFD) allows the user to monitor the external clock or crystal oscillator signal provided by the external oscillator (XOSC32K). The CFD detects failing operation of the XOSC32K clock with reduced latency, and supports switching to a safe clock source in case of clock failure. The user can also switch from the safe clock back to XOSC32K in case of recovery. The safe clock is derived from the OSCULP32K oscillator with a configurable prescaler. This supports configuring the safe clock in order to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested by a peripheral. See the Sleep Behavior table above when this is the case.

The user interface registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can optionally trigger an interrupt or an event when a failure is detected.

**Clock Failure Detection**

The CFD is reset only at power-on (POR). The CFD does not monitor the XOSC32K clock when the oscillator is disabled (XOSC32K.ENABLE=0).

Before starting CFD operation, the user must start and enable the safe clock source (OSCULP32K oscillator).

CFD operation is started by writing a '1' to the CFD Enable bit in the External Oscillator Control register (CFDCTRL.CFDEN). After starting or restarting the XOSC32K, the CFD does not detect failure until the start-up time has elapsed. The start-up time is configured by the Oscillator Start-Up


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Time in the External Multipurpose Crystal Oscillator Control register (XOSC32K.STARTUP). Once the XOSC32K Start-Up Time is elapsed, the XOSC32K clock is constantly monitored.

During a period of 4 safe clocks (monitor period), the CFD watches for a clock activity from the XOSC32K. There must be at least one rising and one falling XOSC32K clock edge during 4 safe clock periods to meet non-failure conditions. If no or insufficient activity is detected, the failure status is asserted: The Clock Failure Detector status bit in the Status register (STATUS.XOSC32KFAIL) and the Clock Failure Detector interrupt flag bit in the Interrupt Flag register (INTFLAG.XOSC32KFAIL) are set. If the XOSC32KFAIL bit in the Interrupt Enable Set register (INTENSET.XOSC32KFAIL) is set, an interrupt is generated as well. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated, too.

After a clock failure was issued the monitoring of the XOSC32K clock is continued, and the Clock Failure Detector status bit in the Status register (STATUS.XOSC32KFAIL) reflects the current XOSC32K activity.

#### Clock Switch

When a clock failure is detected, the XOSC32K clock is replaced by the safe clock in order to maintain an active clock during the XOSC32K clock failure. The safe clock source is the OSCULP32K oscillator clock. Both 32KHz and 1KHz outputs of the XOSC32K are replaced by the respective OSCULP32K 32KHz and 1KHz outputs. The safe clock source can be scaled down by a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC32K clock is switched to the safe clock, the Clock Switch bit in the Status register (STATUS.XOSC32KSW) is set.

When the CFD has switched to the safe clock, the XOSC32K is not disabled. If desired, the application must take the necessary actions to disable the oscillator. The application must also take the necessary actions to configure the system clocks to continue normal operations. In the case the application can recover the XOSC32K, the application can switch back to the XOSC32K clock by writing a '1' to Switch Back Enable bit in the Clock Failure Control register (CFDCTRL.SWBACK). Once the XOSC32K clock is switched back, the Switch Back bit (CFDCTRL.SWBACK) is cleared by hardware.

#### Prescaler

The CFD has an internal configurable prescaler to generate the safe clock from the OSCULP32K oscillator. The prescaler size allows to scale down the OSCULP32K oscillator so the safe clock frequency is not higher than the XOSC32K clock frequency monitored by the CFD. The maximum division factor is 2.

The prescaler is applied on both outputs (32KHz and 1KHz) of the safe clock.

**Example 29-1.** Example

For an external crystal oscillator at 32KHz and the OSCULP32K frequency is 32KHz, the XOSC32K.CFDPRESC should be set to 0 for a safe clock of equal frequency.

#### Event

If the Event Output Enable bit in the Event Control register (EVCTRL.CFDEO) is set, the CFD clock failure will be output on the Event Output. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

#### Sleep Mode

The CFD is halted depending on configuration of the XOSC32K and the peripheral clock request. For further details, refer to the Sleep Behavior table above. The CFD interrupt can be used to wake up the device from sleep modes.

### 29.6.4. 32 kHz Ultra Low-Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a tunable, low-speed, and ultra low-power clock source.
**Note:** The OSCULP32K is factory-calibrated under typical voltage and temperature conditions.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The OSCULP32K is enabled by default after a Power-on Reset (POR), and will always run except during POR.

Users can lock the OSCULP32K configuration by setting the Write Lock bit in the 32 kHz Ultra Low-Power Internal Oscillator Control register (OSCULP32K.WRTLOCK = 1). If set, the OSCULP32K configuration is locked until POR is detected.

The OSCULP32K can be used as a source for Generic Clock Generators (GCLK) or for the Real-Time Counter (RTC). To ensure proper operation, the GCLK or RTC modules must be disabled before the clock selection is changed.

**Related Links**

RTC – Real-Time Counter
Real-Time Counter Clock Selection
GCLK - Generic Clock Controller

### 29.6.5. Watchdog Timer Clock Selection

The Watchdog Timer (WDT) uses the internal 1.024 kHz OSCULP32K output clock. This clock is running all the time and internally enabled when requested by the WDT module.

**Related Links**

WDT – Watchdog Timer

### 29.6.6. Real-Time Counter Clock Selection

Before enabling the RTC module, the RTC clock must be selected first. All oscillator outputs are valid as RTC clock. The selection is done in the RTC Control register (RTCCTRL). To ensure a proper operation, it is highly recommended to disable the RTC module first, before the RTC clock source selection is changed.

**Related Links**

RTC – Real-Time Counter

### 29.6.7. Interrupts

The OSC32KCTRL has the following interrupt sources:

- XOSC32KRDY - 32KHz Crystal Oscillator Ready: A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected
- XOSC32KFAIL - Clock Failure Detector: A 0-to-1 transition on the STATUS.XOSC32KFAIL bit is detected

All these interrupts are synchronous wake-up source.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be enabled individually by setting the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the OSC32KCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSC32KCTRL has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present. Refer to the INTFLAG register for details.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**

PM – Power Manager
Nested Vector Interrupt Controller

### 29.6.8. Events

The CFD can generate the following output event:

- Clock Failure Detector (XOSC32KFAIL): Generated when the Clock Failure Detector status bit is set in the Status register (STATUS.XOSC32KFAIL). The CFD event is not generated when the Clock Switch bit (STATUS.SWBACK) in the Status register is set.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.CFDEO) enables the CFD output event. Writing a '0' to this bit disables the CFD output event. Refer to the Event System chapter for details on configuring the event system.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 29.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | INTENCLR | 7:0 | | | | | | XOSC32KFAIL | | XOSC32KRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | INTENSET | 7:0 | | | | | | XOSC32KFAIL | | XOSC32KRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | INTFLAG | 7:0 | | | | | | XOSC32KFAIL | | XOSC32KRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | STATUS | 7:0 | | | | | XOSC32KSW | XOSC32KFAIL | | XOSC32KRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | RTCCTRL | 7:0 | | | | | | RTCSEL[2:0] | | |
| 0x11 ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | XOSC32K | 7:0 | ONDEMAND | RUNSTDBY | | EN1K | EN32K | XTALEN | ENABLE | |
| | | 15:8 | | CGM[1:0] | | WRTLOCK | | STARTUP[2:0] | | |
| 0x16 | CFDCTRL | 7:0 | | | | | | CFDPRESC | SWBACK | CFDEN |
| 0x17 | EVCTRL | 7:0 | | | | | | | | CFDEO |
| 0x18 ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | OSCULP32K | 7:0 | | | | | | EN1K | EN32K | |
| | | 15:8 | WRTLOCK | | Reserved[5:0] | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 29.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register and the 8-bit halves of a 16-bit register can be accessed directly.

All registers with write-access can be write-protected optionally by the peripheral access controller (PAC). Optional Write-Protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write-Protection" property in the register description. Write-protection does not apply to accesses through an external debugger.

**Related Links**

PAC - Peripheral Access Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.1. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | XOSC32KFAIL | | XOSC32KRDY |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC32K Clock Failure Interrupt Enable bit, which disables the XOSC32K Clock Failure interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC32K Clock Failure Detection is disabled. |
| 1 | The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set. |

**Bit 0 – XOSC32KRDY** XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC32K Ready interrupt is disabled. |
| 1 | The XOSC32K Ready interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.2. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | XOSC32KFAIL | | XOSC32KRDY |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the XOSC32K Clock Failure Interrupt Enable bit, which enables the XOSC32K Clock Failure interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC32K Clock Failure Detection is disabled. |
| 1 | The XOSC32K Clock Failure Detection is enabled. An interrupt request will be generated when the XOSC32K Clock Failure Detection interrupt flag is set. |

**Bit 0 – XOSC32KRDY** XOSC32K Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the XOSC32K Ready Interrupt Enable bit, which enables the XOSC32K Ready interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC32K Ready interrupt is disabled. |
| 1 | The XOSC32K Ready interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.3. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | XOSC32KFAIL | | XOSC32KRDY |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector

This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC32K Clock Failure Detection bit in the Status register (STATUS.XOSC32KFAIL) and will generate an interrupt request if INTENSET.XOSC32KFAIL is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC32K Clock Failure Detection flag.

**Bit 0 – XOSC32KRDY** XOSC32K Ready

This flag is cleared by writing a '1' to it.
This flag is set by a zero-to-one transition of the XOSC32K Ready bit in the Status register (STATUS.XOSC32KRDY), and will generate an interrupt request if INTENSET.XOSC32KRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the XOSC32K Ready interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.4. Status

**Name:** STATUS
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | XOSC32KSW | XOSC32KFAIL | | XOSC32KRDY |
| Access | | | | | R | R | | R |
| Reset | | | | | 0 | 0 | | 0 |

**Bit 3 – XOSC32KSW** XOSC32K Clock Switch

| Value | Description |
|---|---|
| 0 | XOSC32K is not switched and provided the crystal oscillator. |
| 1 | XOSC32K is switched to be provided by the safe clock. |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector

| Value | Description |
|---|---|
| 0 | XOSC32K is passing failure detection. |
| 1 | XOSC32K is not passing failure detection. |

**Bit 0 – XOSC32KRDY** XOSC32K Ready

| Value | Description |
|---|---|
| 0 | XOSC32K is not ready. |
| 1 | XOSC32K is stable and ready to be used as a clock source. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.5. RTC Clock Selection Control

**Name:** RTCCTRL
**Offset:** 0x10
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | RTCSEL[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 2:0 – RTCSEL[2:0]** RTC Clock Selection
These bits select the source for the RTC.

| Value | Name | Description |
|---|---|---|
| 0x0 | ULP1K | 1.024 kHz from 32 KHz internal ULP oscillator |
| 0x1 | ULP32K | 32.768 kHz from 32 KHz internal ULP oscillator |
| 0x2, 0x3 | Reserved | - |
| 0x4 | XOSC1K | 1.024 kHz from 32 KHz external oscillator |
| 0x5 | XOSC32K | 32.768 kHz from 32 KHz external crystal oscillator |
| 0x6 | Reserved | - |
| 0x7 | Reserved | - |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.6. 32 KHz External Crystal Oscillator (XOSC32K) Control

**Name:** XOSC32K
**Offset:** 0x14
**Reset:** 0x2080
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | CGM[1:0] | | WRTLOCK | | STARTUP[2:0] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 1 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | EN1K | EN32K | XTALEN | ENABLE | |
| Access | R/W | R/W | | R/W | R/W | R/W | R/W | |
| Reset | 1 | 0 | | 0 | 0 | 0 | 0 | |

**Bits 14:13 – CGM[1:0]** Control Gain Mode
These bits control the gain of the external crystal oscillator.

| Value | Name | Description |
|---|---|---|
| 0x1 | XT | Standard mode |
| 0x2 | HS | High Speed mode |

**Bit 12 – WRTLOCK** Write Lock
This bit locks the XOSC32K register for future writes, effectively freezing the XOSC32K configuration.

| Value | Description |
|---|---|
| 0 | The XOSC32K configuration is not locked. |
| 1 | The XOSC32K configuration is locked. |

**Bits 10:8 – STARTUP[2:0]** Oscillator Start-Up Time
These bits select the start-up time for the oscillator.
The OSCULP32K oscillator is used to clock the start-up counter.

**Table 29-2.** Start-Up Time for 32 kHz External Crystal Oscillator

| STARTUP[2:0] | Number of OSCULP32K Clock Cycles | Number of XOSC32K Clock Cycles | Approximate Equivalent Time [ms] |
|---|---|---|---|
| 0x0 | 2048 | 3 | 62.592 |
| 0x1 | 4096 | 3 | 125.092 |
| 0x2 | 16384 | 3 | 500.092 |
| 0x3 | 32768 | 3 | 1000.0092 |
| 0x4 | 65536 | 3 | 2000.0092 |
| 0x5 | 131072 | 3 | 4000.0092 |
| 0x6 | 262144 | 3 | 8000.0092 |
| 0x7 | - | - | Reserved |

**Notes:**
1. Actual Start-Up time is 1 OSCULP32K cycle + 3 XOSC32K cycles.
2. The given time assumes an XTAL frequency of 32.768 kHz.

**Bit 7 – ONDEMAND** On Demand Control
This bit controls how the XOSC32K behaves when a peripheral clock request is detected. For details, refer to Table 29-1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 6 – RUNSTDBY** Run in Standby

This bit controls how the XOSC32K behaves during standby sleep mode. For details, refer to Table 29-1.

**Bit 4 – EN1K** 1 kHz Output Enable

| Value | Description |
|---|---|
| 0 | The 1.024 kHz output is disabled. |
| 1 | The 1.024 kHz output is enabled. |

**Bit 3 – EN32K** 32 kHz Output Enable

| Value | Description |
|---|---|
| 0 | The 32.768 kHz output is disabled. |
| 1 | The 32.768 kHz output is enabled. |

**Bit 2 – XTALEN** Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator.

| Value | Description |
|---|---|
| 0 | External clock connected on XIN32. XOUT32 can be used as general-purpose I/O. |
| 1 | Crystal connected to XIN32/XOUT32. |

**Bit 1 – ENABLE** Oscillator Enable

| Value | Description |
|---|---|
| 0 | The oscillator is disabled. |
| 1 | The oscillator is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 29.8.7. Clock Failure Detector Control

**Name:** CFDCTRL
**Offset:** 0x16
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CFDPRESC | SWBACK | CFDEN |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – CFDPRESC** Clock Failure Detector Prescaler
This bit selects the prescaler for the Clock Failure Detector.

| Value | Description |
|---|---|
| 0 | The CFD safe clock frequency is the OSCULP32K frequency |
| 1 | The CFD safe clock frequency is the OSCULP32K frequency divided by 2 |

**Bit 1 – SWBACK** Clock Switch Back
This bit clontrols the XOSC32K output switch back to the external clock or crystal scillator in case of clock recovery.

| Value | Description |
|---|---|
| 0 | The clock switch is disabled. |
| 1 | The clock switch is enabled. This bit is reset when the XOSC32K output is switched back to the external clock or crystal oscillator. |

**Bit 0 – CFDEN** Clock Failure Detector Enable
This bit selects the Clock Failure Detector state.

| Value | Description |
|---|---|
| 0 | The CFD is disabled. |
| 1 | The CFD is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.8. Event Control

**Name:** EVCTRL
**Offset:** 0x17
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CFDEO |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – CFDEO** Clock Failure Detector Event Out Enable
This bit controls whether the Clock Failure Detector event output is enabled and an event will be generated when the CFD detects a clock failure.

| Value | Description |
|---|---|
| 0 | Clock Failure Detector Event output is disabled, no event will be generated. |
| 1 | Clock Failure Detector Event output is enabled, an event will be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 29.8.9. 32KHz Ultra Low Power Internal Oscillator (OSCULP32K) Control

**Name:** OSCULP32K
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WRTLOCK | | Reserved[5:0] | | | | | |
| Access | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | x | x | x | x | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | EN1K | EN32K | |
| Access | | | | | | R/W | R/W | |
| Reset | | | | | | 1 | 1 | |

**Bit 15 – WRTLOCK** Write Lock
This bit locks the OSCULP32K register for future writes to fix the OSCULP32K configuration.

| Value | Description |
|---|---|
| 0 | The OSCULP32K configuration is not locked. |
| 1 | The OSCULP32K configuration is locked. |

**Bits 13:8 – Reserved[5:0]**
These bits must not be changed to ensure correct OSCULP32K behavior.

**Bit 2 – EN1K** 1kHz Output Enable

| Value | Description |
|---|---|
| 0 | The 1kHz output is disabled |
| 1 | The 1kHz output is enabled. |

**Bit 1 – EN32K**

| Value | Description |
|---|---|
| 0 | The 32kHz output is disabled. |
| 1 | The 32kHz output is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 30. FREQM – Frequency Meter

## 30.1. Overview

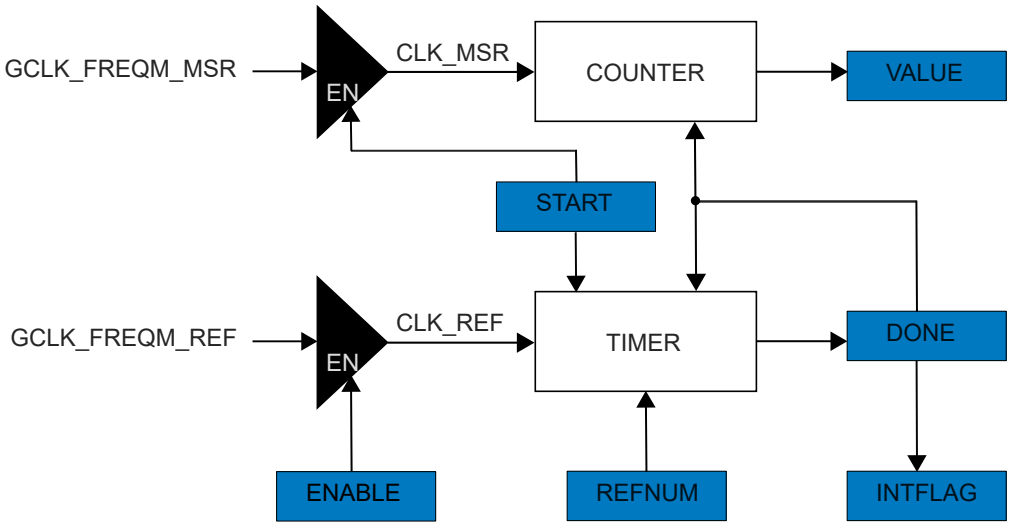
The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock by comparing it to a known reference clock.

## 30.2. Features

- Ratio can be measured with 24-bit accuracy
- Accurately measures the frequency of an input clock with respect to a reference clock
- Reference clock can be selected from the available GCLK_FREQM_REF sources
- Measured clock can be selected from the available GCLK_FREQM_MSR sources

## 30.3. Block Diagram

**Figure 30-1.** FREQM Block Diagram




## 30.4. Signal Description

Not applicable.

## 30.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 30.5.1. I/O Lines

The GCLK I/O lines (GCLK_IO[7:0]) can be used as measurement or reference clock sources. This requires the I/O pins to be configured.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.5.2. Power Management

The FREQM will continue to operate in idle sleep mode where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep mode. Refer to the Power Manager chapter for details on the different sleep modes.

**Related Links**

PM – Power Manager

### 30.5.3. Clocks

The clock for the FREQM bus interface (CLK_FREQM_APB) is enabled and disabled by the Main Clock Controller, the default state of CLK_APB_FREQM can be found in Peripheral Clock Masking.

Two generic clocks are used by the FREQM: Reference Clock (GCLK_FREQM_REF) and Measurement Clock (GCLK_FREQM_MSR).

GCLK_FREQM_REF is required to clock the internal reference timer, which acts as the frequency reference.

GCLK_FREQM_MSR is required to clock a ripple counter for frequency measurement. These clocks must be configured and enabled in the generic clock controller before using the FREQM.

**Related Links**

MCLK – Main Clock
Peripheral Clock Masking
GCLK - Generic Clock Controller

### 30.5.4. DMA

Not applicable.

### 30.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using FREQM interrupt requires the interrupt controller to be configured first.

**Related Links**

Interrupt Line Mapping

### 30.5.6. Events

Not applicable

### 30.5.7. Debug Operation

When the CPU is halted in debug mode the FREQM continues its normal operation. The FREQM cannot be halted when the CPU is halted in debug mode. If the FREQM is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

### 30.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except the following registers:

- Control B register (CTRLB)
- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
PAC - Peripheral Access Controller

## 30.6. Functional Description

### 30.6.1. Principle of Operation

FREQM counts the number of periods of the measured clock (GCLK_FREQM_MSR) with respect to the reference clock (GCLK_FREQM_REF). The measurement is done for a period of REFNUM/$f_{CLK_REF}$ and stored in the Value register (VALUE.VALUE). REFNUM is the number of Reference clock cycles selected in the Configuration A register (CFGA.REFNUM).

The frequency of the measured clock, $f_{CLK_MSR}$, is calculated by

$$f_{CLK_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK_REF}$$

### 30.6.2. Basic Operation

#### 30.6.2.1. Initialization

Before enabling FREQM, the device and peripheral must be configured:

- Each of the generic clocks (GCLK_FREQM_REF and GCLK_FREQM_MSR) must be configured and enabled.
- 

> **Important:** The reference clock must be slower than the measurement clock.

- Write the number of Reference clock cycles for which the measurement is to be done in the Configuration A register (CFGA.REFNUM). This must be a non-zero number.

The following register is enable-protected, meaning that it can only be written when the FREQM is disabled (CTRLA.ENABLE=0):

- Configuration A register (CFGA)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

**Related Links**
GCLK - Generic Clock Controller

#### 30.6.2.2. Enabling, Disabling and Resetting

The FREQM is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The peripheral is disabled by writing CTRLA.ENABLE=0.

The FREQM is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). On software reset, all registers in the FREQM will be reset to their initial state, and the FREQM will be disabled.

Then ENABLE and SWRST bits are write-synchronized.

**Related Links**
Synchronization

#### 30.6.2.3. Measurement

In the Configuration A register, the Number of Reference Clock Cycles field (CFGA.REFNUM) selects the duration of the measurement. The measurement is given in number of GCLK_FREQM_REF periods.
**Note:** The REFNUM field must be written before the FREQM is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

After the FREQM is enabled, writing a '1' to the START bit in the Control B register (CTRLB.START) starts the measurement. The BUSY bit in Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

There is also an interrupt request for Measurement Done: When the Measurement Done bit in Interrupt Enable Set register (INTENSET.DONE) is '1' and a measurement is finished, the Measurement Done bit in the Interrupt Flag Status and Clear register (INTFLAG.DONE) will be set and an interrupt request is generated.

The result of the measurement can be read from the Value register (VALUE.VALUE). The frequency of the measured clock GCLK_FREQM_MSR is then:

$$f_{CLK_MSR} = \left(\frac{VALUE}{REFNUM}\right) f_{CLK_REF}$$

**Note:** In order to make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) should be checked.

In case an overflow condition occurred, indicated by the Overflow bit in the STATUS register (STATUS.OVF), either the number of reference clock cycles must be reduced (CFGA.REFNUM), or a faster reference clock must be configured. Once the configuration is adjusted, clear the overflow status by writing a '1' to STATUS.OVF. Then another measurement can be started by writing a '1' to CTRLB.START.

### 30.6.3. DMA Operation

Not applicable.

### 30.6.4. Interrupts

The FREQM has one interrupt source:

- DONE: A frequency measurement is done.

The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the FREQM is reset. See INTFLAG for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

This interrupt is a synchronous wake-up source.

Note that interrupts must be globally enabled for interrupt requests to be generated.

### 30.6.5. Events

Not applicable.

### 30.6.6. Sleep Mode Operation

The FREQM will continue to operate in idle sleep mode where the selected source clock is running. The FREQM's interrupts can be used to wake up the device from idle sleep mode.

For lowest chip power consumption in sleep modes, FREQM should be disabled before entering a sleep mode.

**Related Links**

PM – Power Manager


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits and registers are write-synchronized:

- Software Reset bit in Control A register (CTRLA.SWRST)
- Enable bit in Control A register (CTRLA.ENABLE)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

**Related Links**

Register Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 30.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | | ENABLE | SWRST |
| 0x01 | CTRLB | 7:0 | | | | | | | | START |
| 0x02 | CFGA | 7:0 | REFNUM[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| 0x04 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | | | | | | | | DONE |
| 0x09 | INTENSET | 7:0 | | | | | | | | DONE |
| 0x0A | INTFLAG | 7:0 | | | | | | | | DONE |
| 0x0B | STATUS | 7:0 | | | | | | | OVF | BUSY |
| 0x0C | SYNCBUSY | 7:0 | | | | | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | VALUE | 7:0 | VALUE[7:0] | | | | | | | |
| | | 15:8 | VALUE[15:8] | | | | | | | |
| | | 23:16 | VALUE[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |

## 30.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ENABLE** Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the FREQM to their initial state, and the FREQM will be disabled. Writing a '1' to this bit will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | There is no ongoing Reset operation. |
| 1 | The Reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.2. Control B

**Name:** CTRLB
**Offset:** 0x01
**Reset:** 0x00
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | START |
| Access | | | | | | | | W |
| Reset | | | | | | | | 0 |

**Bit 0 – START** Start Measurement

| Value | Description |
|---|---|
| 0 | Writing a '0' has no effect. |
| 1 | Writing a '1' starts a measurement. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.3. Configuration A

**Name:** CFGA
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | REFNUM[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – REFNUM[7:0]** Number of Reference Clock Cycles
Selects the duration of a measurement in number of CLK_FREQM_REF cycles. This must be a non-zero value, i.e. 0x01 (one cycle) to 0xFF (255 cycles).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DONE |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DONE** Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Measurement Done Interrupt Enable bit, which disables the Measurement Done interrupt.

| Value | Description |
|---|---|
| 0 | The Measurement Done interrupt is disabled. |
| 1 | The Measurement Done interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DONE |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DONE** Measurement Done Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Measurement Done Interrupt Enable bit, which enables the Measurement Done interrupt.

| Value | Description |
|---|---|
| 0 | The Measurement Done interrupt is disabled. |
| 1 | The Measurement Done interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DONE |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DONE** Mesurement Done
This flag is cleared by writing a '1' to it.
This flag is set when the STATUS.BUSY bit has a one-to-zero transition.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the DONE interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 30.8.7. Status

**Name:** STATUS
**Offset:** 0x0B
**Reset:** 0x00
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | OVF | BUSY |
| Access | | | | | | | R/W | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – OVF** Sticky Count Value Overflow
This bit is cleared by writing a '1' to it.
This bit is set when an overflow condition occurs to the value counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the OVF status.

**Bit 0 – BUSY** FREQM Status

| Value | Description |
|---|---|
| 0 | No ongoing frequency measurement. |
| 1 | Frequency measurement is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.8. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ENABLE** Enable
This bit is cleared when the synchronization of CTRLA.ENABLE is complete.
This bit is set when the synchronization of CTRLA.ENABLE is started.

**Bit 0 – SWRST** Synchronization Busy
This bit is cleared when the synchronization of CTRLA.SWRST is complete.
This bit is set when the synchronization of CTRLA.SWRST is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 30.8.9. Value

**Name:** VALUE
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | VALUE[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | VALUE[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VALUE[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:0 – VALUE[23:0]** Measurement Value
Result from measurement.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 31. Event System (EVSYS)

## 31.1. Overview

The Event System (EVSYS) allows autonomous, low-latency and configurable communication between peripherals.

Several peripherals can be configured to generate and/or respond to signals known as events. The exact condition to generate an event, or the action taken upon receiving an event, is specific to each peripheral. Peripherals that respond to events are called event users. Peripherals that generate events are called event generators. A peripheral can have one or more event generators and can have one or more event users.

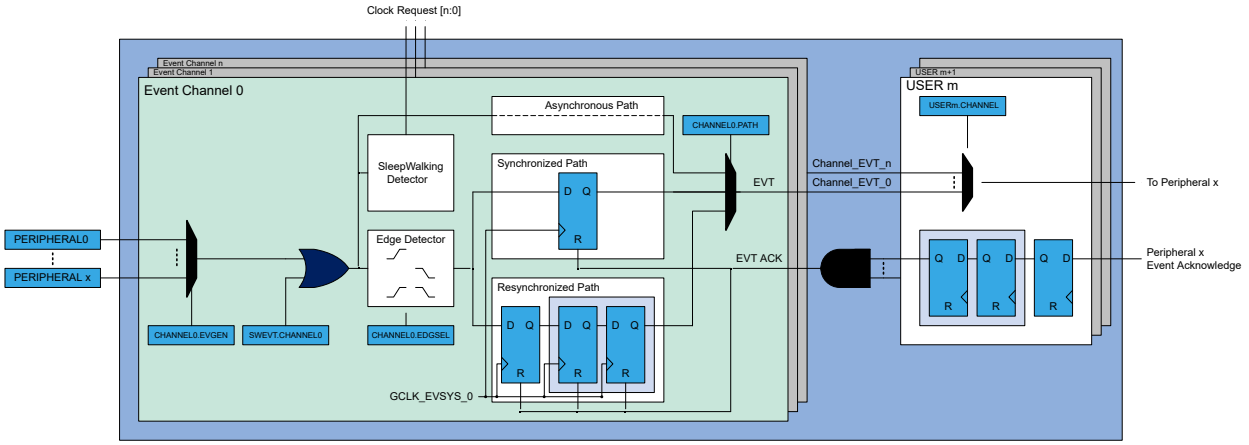
Communication is made without CPU intervention and without consuming system resources such as bus or RAM bandwidth. This reduces the load on the CPU and other system resources, compared to a traditional interrupt-based system.

## 31.2. Features

- 32 configurable event channels:
  - All channels can be connected to any event generator
  - All channels provide a pure asynchronous path
  - 12 channels (CHANNEL0 to CHANNEL11) provide a resynchronized or synchronous path using their dedicated generic clock (GCLK_EVSYS_CHANNEL_n)
- 119 event generators.
- 67 event users.
- Configurable edge detector.
- Peripherals can be event generators, event users, or both.
- SleepWalking and interrupt for operation in sleep modes.
- Software event generation.
- Each event user can choose which channel to respond to.
- Optional Static or Round-Robin interrupt priority arbitration.

## 31.3. Block Diagram

**Figure 31-1.** Event System Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 31.4. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 31.4.1. I/O Lines

Not applicable.

### 31.4.2. Power Management

The EVSYS can be used to wake up the CPU from all sleep modes (except BACKUP and OFF Mode), even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM – Power Manager* for details on the different sleep modes.

Although the clock for the EVSYS is stopped, the device still can wake up the EVSYS clock. Some event generators can generate an event when their clocks are stopped. The generic clock for the channel (GCLK_EVSYS_CHANNEL_n) will be restarted if that channel uses a synchronized path or a resynchronized path. It does not need to wake the system from sleep.

**Important:** This generic clock only applies to channels which can be configured as synchronous or resynchronized.

**Related Links**

PM – Power Manager

### 31.4.3. Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_EVSYS_APB can be found in *Peripheral Clock Masking*.

Each EVSYS channel which can be configured as synchronous or resynchronized has a dedicated generic clock (GCLK_EVSYS_CHANNEL_n). These are used for event detection and propagation for each channel. These clocks must be configured and enabled in the generic clock controller before using the EVSYS. Refer to *GCLK - Generic Clock Controller* for details.

**Important:** Only EVSYS channel 0 to 11 can be configured as synchronous or resynchronized.

**Related Links**

Peripheral Clock Masking

GCLK - Generic Clock Controller

### 31.4.4. DMA

Not applicable.

### 31.4.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the EVSYS interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller

### 31.4.6. Events

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.4.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging.

### 31.4.8. Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Channel Pending Interrupt (INTPEND)
- Channel n Interrupt Flag Status and Clear (CHINTFLAGn)

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

### 31.4.9. Analog Connections

Not applicable.

## 31.5. Functional Description

### 31.5.1. Principle of Operation

The Event System consists of several channels which route the internal events from peripherals (generators) to other internal peripherals or I/O pins (users). Each event generator can be selected as source for multiple channels, but a channel cannot be set to use multiple event generators at the same time.

A channel path can be configured in asynchronous, synchronous or resynchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using synchronous or resynchronized path, the Event System includes options to transfer events to users when rising, falling or both edges are detected on event generators.

For further details, refer to the Channel Path section of this chapter.

**Related Links**

Channel Path

### 31.5.2. Basic Operation

#### 31.5.2.1. Initialization

Before enabling event routing within the system, the Event Users Multiplexer and Event Channels must be selected in the Event System (EVSYS), and the two peripherals that generate and use the event must be configured. Follow these steps to configure the event:

1. In the peripheral generating the event, enable output of event by writing a '1' to the respective Event Output Enable bit ("EO") in the peripheral's Event Control register, for example, AC.EVCTRL.WINEO0 or RTC.EVCTRL.OVFEO.
2. Configure the EVSYS:
   a. Configure the Event User multiplexer by writing the respective EVSYS.USERm register, refer to User Multiplexer Setup.
   b. Configure the Event Channel by writing the respective EVSYS.CHANNELn register, refer to Event System Channel.
3. Configure the action to be executed by the event user peripheral by writing to the Event Action bits (EVACT) in the respective Event control register, for example, TC.EVCTRL.EVACT, PDEC.EVCTRL.EVACT.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** This step is not applicable for all the peripherals.

4. In the event user peripheral, enable event input by writing a '1' to the respective Event Input Enable bit ("EI") in the peripheral's Event Control register, for example, AC.EVCTRL.IVEI0, ADC.EVCTRL.STARTEI.

#### 31.5.2.2. Enabling, Disabling, and Resetting

The EVSYS is always enabled.

The EVSYS is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the EVSYS will be reset to their initial state and all ongoing events will be canceled.

Refer to CTRLA.SWRST register for details.

#### 31.5.2.3. User Multiplexer Setup

The user multiplexer defines the channel to be connected to which event user. Each user multiplexer is dedicated to one event user. A user multiplexer receives all event channels output and must be configured to select one of these channels, as shown in Block Diagram section. The channel is selected with the Channel bit group in the User register (USERm.CHANNEL).

The user multiplexer must always be configured before the channel. A list of all available event users is found in the User (USERm) register description.

**Related Links**
Block Diagram

#### 31.5.2.4. Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event could be synchronized, resynchronized or asynchronously sent to the users. When synchronization or resynchronization is required, the channel includes an internal edge detector, allowing the Event System to generate internal events when rising, falling or both edges are detected on the selected event generator.

An event channel is able to generate internal events for the specific software commands. A channel block diagram is shown in *Block Diagram* section.

**Related Links**
Block Diagram

#### 31.5.2.5. Event Generators

Each event channel can receive the events form all event generators. All event generators are listed in the Event Generator bit field in the Channel n register (CHANNELn.EVGEN). For details on event generation, refer to the corresponding module chapter. The channel event generator is selected by the Event Generator bit group in the Channel register (CHANNELn.EVGEN). By default, the channels are not connected to any event generators (ie, CHANNELn.EVGEN = 0)

#### 31.5.2.6. Channel Path

There are different ways to propagate the event from an event generator:

- Asynchronous path
- Synchronous path
- Resynchronized path

The path is decided by writing to the Path Selection bit group of the Channel register (CHANNELn.PATH).

**Asynchronous Path**
When using the asynchronous path, the events are propagated from the event generator to the event user without intervention from the Event System. The GCLK for this channel


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

(GCLK_EVSYS_CHANNEL_n) is not mandatory, meaning that an event will be propagated to the user without any clock latency.

When the asynchronous path is selected, the channel cannot generate any interrupts, and the Channel x Status register (CHSTATUSx) is always zero. The edge detection is not required and must be disabled by software. Each peripheral event user has to select which event edge must trigger internal actions. For further details, refer to each peripheral chapter description.

**Synchronous Path**

The synchronous path should be used when the event generator and the event channel share the same generator for the generic clock. If they do not share the same clock, a logic change from the event generator to the event channel might not be detected in the channel, which means that the event will not be propagated to the event user.

When using the synchronous path, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

**Resynchronized Path**

The resynchronized path are used when the event generator and the event channel do not share the same generator for the generic clock. When the resynchronized path is used, resynchronization of the event from the event generator is done in the channel.

When the resynchronized path is used, the channel is able to generate interrupts. The channel status bits in the Channel Status register (CHSTATUS) are also updated and available for use.

#### 31.5.2.7. Edge Detection

When synchronous or resynchronized paths are used, edge detection must be enabled. The event system can execute edge detection in three different ways:

- Generate an event only on the rising edge
- Generate an event only on the falling edge
- Generate an event on rising and falling edges.

Edge detection is selected by writing to the Edge Selection bit group of the Channel register (CHANNELn.EDGSEL).

#### 31.5.2.8. Event Latency

The latency from event generator to event user depends on the channel's configuration:

- Asynchronous Path: The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.
- Synchronous Path: The maximum routing latency of an external event is one GCLK_EVSYS_CHANNEL_n clock cycle.
- Resynchronized Path: The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_n clock cycles.

The maximum propagation latency of a user event to the peripheral clock core domain is three peripheral clock cycles.

The event generators, event channel and event user clocks ratio must be selected in relation with the internal event latency constraints. Events propagation or event actions in peripherals may be lost if the clock setup violates the internal latencies.

#### 31.5.2.9. The Overrun Channel n Interrupt

The Overrun Channel n Interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.OVR) will be set, and the optional interrupt will be generated in the following cases:

- One or more event users on channel n is not ready when there is a new event


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- An event occurs when the previous event on channel m has not been handled by all event users connected to that channel

The flag will only be set when using synchronous or resynchronized paths. In the case of asynchronous path, the CHINTFLAGn.OVR is always read as zero.

### 31.5.2.10. The Event Detected Channel n Interrupt

The Event Detected Channel n Interrupt flag in the Interrupt Flag Status and Clear register (CHINTFLAGn.EVD) is set when an event coming from the event generator configured on channel n is detected.

The flag will only be set when using a synchronous or resynchronized path. In the case of an asynchronous path, the CHINTFLAGn.EVD is always zero.

### 31.5.2.11. Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a synchronous or resynchronized path. There are two different status bits in CHSTATUS for each of the available channels:

- The CHSTATUSn.BUSYCH bit will be set when an event on the corresponding channel n has not been handled by all event users connected to that channel.
- The CHSTATUSn.RDYUSR bit will be set when all event users connected to the corresponding channel are ready to handle incoming events on that channel.

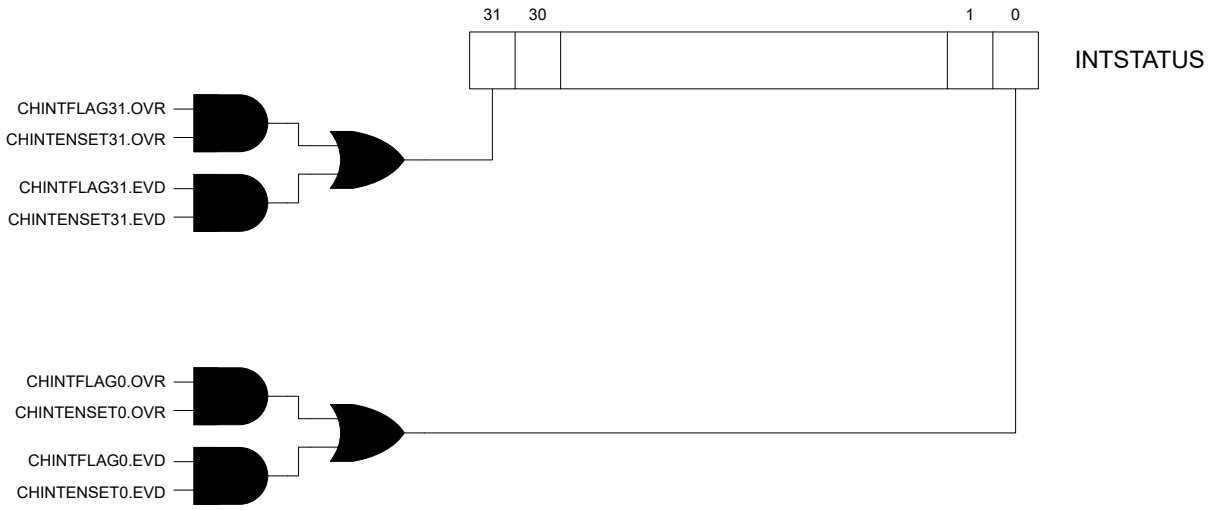
### 31.5.2.12. Software Event

A software event can be initiated on a channel by writing a '1' to the Software Event bit in the Channel register (SWEVT.CHANNELn). Then the software event can be serviced as any event generator; i.e., when a bit is set to '1', the corresponding event will be generated on the respective channel.

### 31.5.2.13. Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as shown below.

**Figure 31-2.** Interrupt Status Register

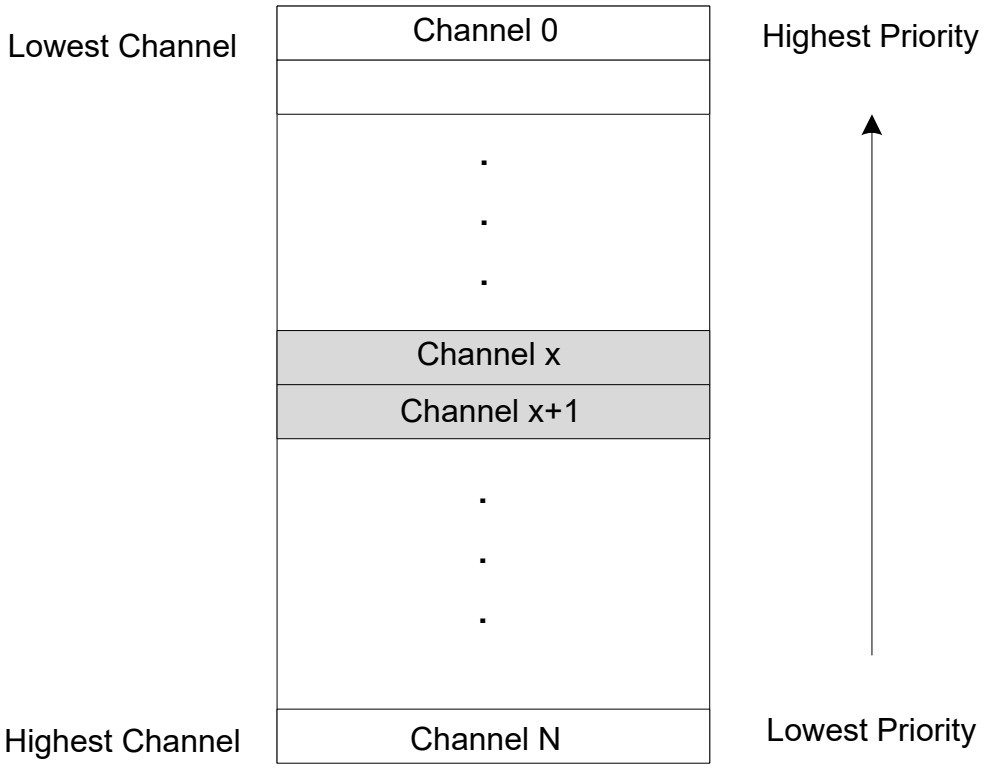



The Event System can arbitrate between all channels with pending interrupts. The arbiter can be configured to prioritize statically or dynamically the incoming events. The priority is evaluated each time a new channel has an interrupt pending, or an interrupt has been cleared. The Channel Pending Interrupt register (INTPEND) will provide the channel number with the highest interrupt priority, and the corresponding channel interrupt flags and status bits.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

By default, static arbitration is enabled (PRICTRL.RRENx is '0'), the arbiter will prioritize a low channel number over a high channel number as shown below. When using the status scheme, there is a risk of high channel numbers never being granted access by the arbiter. This can be avoided using a dynamic arbitration scheme.

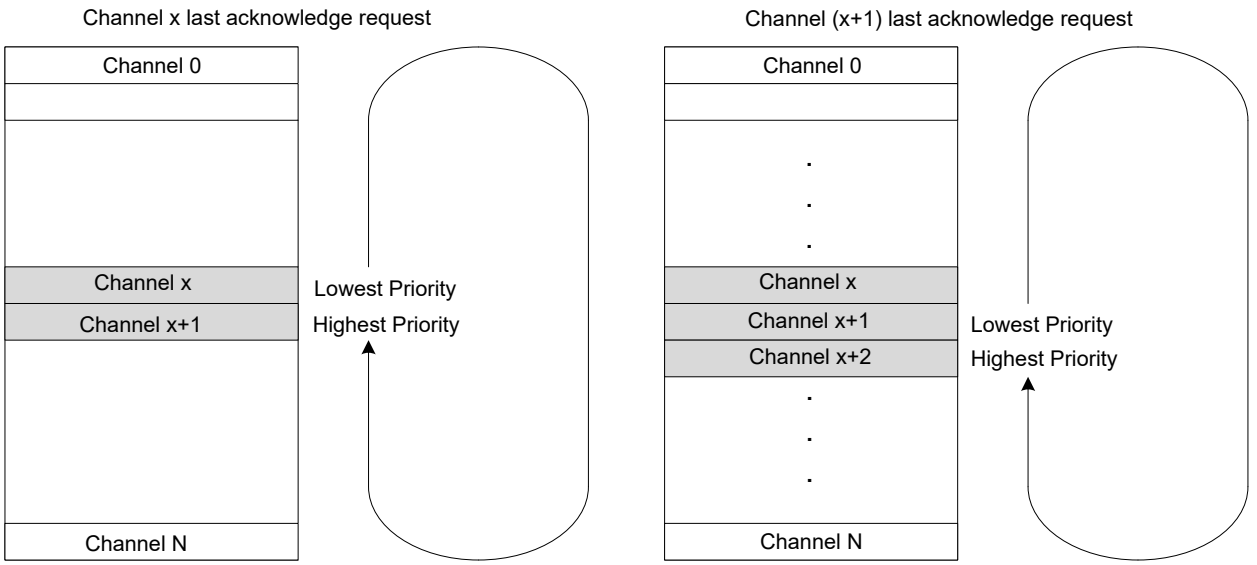
**Figure 31-3.** Static Priority




The dynamic arbitration scheme available in the Event System is round-robin. Round-robin arbitration is enabled by writing PRICTRL.RREN to one. With the round-robin scheme, the channel number of the last channel being granted access will have the lowest priority the next time the arbiter has to grant access to a channel, as shown below. The channel number of the last channel being granted access, will be stored in the Channel Priority Number bit group in the Priority Control register (PRICTRL.PRI).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 31-4.** Round-Robin Scheduling




The Channel Pending Interrupt register (INTPEND) also offers the possibility to indirectly clear the interrupt flags of a specific channel. Writing a flag to one in this register, will clear the corresponding interrupt flag of the channel specified by the INTPEND.ID bits.

### 31.5.3. Interrupts

The EVSYS has the following interrupt sources for each channel:

- Overrun Channel n interrupt (OVR)
- Event Detected Channel n interrupt (EVD)

These interrupts events are asynchronous wake-up sources.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the corresponding Channel n Interrupt Flag Status and Clear (CHINTFLAG) register is set when the interrupt condition occurs.

**Note:** Interrupts must be globally enabled to allow the generation of interrupt requests.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Set (CHINTENSET) register, and disabled by writing a '1' to the corresponding bit in the Channel n Interrupt Enable Clear (CHINTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled or the Event System is reset. All interrupt requests are ORed together on system level to generate one combined interrupt request to the NVIC.

The user must read the Channel Interrupt Status (INTSTATUS) register to identify the channels with pending interrupts, and must read the Channel n Interrupt Flag Status and Clear (CHINTFLAG) register to determine which interrupt condition is present for the corresponding channel. It is also possible to read the Interrupt Pending register (INTPEND), which provides the highest priority channel with pending interrupt and the respective interrupt flags.

### 31.5.4. Sleep Mode Operation

The Event System can generate interrupts to wake up the device from IDLE or STANDBY sleep mode.

To be able to run in standby, the Run in Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in Channel register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

(CHANNELn.ONDEMAND) is set to '1' and the event generator is detected, the event channel will request its clock (GCLK_EVSYS_CHANNEL_n). The event latency for a resynchronized channel path will increase by two GCLK_EVSYS_CHANNEL_n clock (i.e., up to five GCLK_EVSYS_CHANNEL_n clock cycles).

A channel will behave differently in different sleep modes regarding to CHANNELn.RUNSTDBY and CHANNELn.ONDEMAND:

**Table 31-1.** Event Channel Sleep Behavior

| CHANNELn.PATH | CHANNELn. ONDEMAND | CHANNELn. RUNSTDBY | Sleep Behavior |
|---|---|---|---|
| ASYNC | 0 | 0 | Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode. |
| SYNC/RESYNC | 0 | 1 | Run in both IDLE and STANDBY sleep modes. |
| SYNC/RESYNC | 1 | 0 | Only run in IDLE sleep modes if an event must be propagated. Disabled in STANDBY sleep mode. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally. |
| SYNC/RESYNC | 1 | 1 | Run in both IDLE and STANDBY sleep modes. Two GCLK_EVSYS_n latency added in RESYNC path before the event is propagated internally. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 31.6. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | | | SWRST |
| 0x01 ... 0x03 | Reserved | | | | | | | | | |
| 0x04 | SWEVT | 7:0 | CHANNEL7 | CHANNEL6 | CHANNEL5 | CHANNEL4 | CHANNEL3 | CHANNEL2 | CHANNEL1 | CHANNEL0 |
| | | 15:8 | CHANNEL15 | CHANNEL14 | CHANNEL13 | CHANNEL12 | CHANNEL11 | CHANNEL10 | CHANNEL9 | CHANNEL8 |
| | | 23:16 | CHANNEL23 | CHANNEL22 | CHANNEL21 | CHANNEL20 | CHANNEL19 | CHANNEL18 | CHANNEL17 | CHANNEL16 |
| | | 31:24 | CHANNEL31 | CHANNEL30 | CHANNEL29 | CHANNEL28 | CHANNEL27 | CHANNEL26 | CHANNEL25 | CHANNEL24 |
| 0x08 | PRICTRL | 7:0 | RREN | | | | PRI[3:0] | | | |
| 0x09 ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | INTPEND | 7:0 | | | | | ID[3:0] | | | |
| | | 15:8 | BUSY | READY | | | | | EVD | OVR |
| 0x12 ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTSTATUS | 7:0 | CHINT7 | CHINT6 | CHINT5 | CHINT4 | CHINT3 | CHINT2 | CHINT1 | CHINT0 |
| | | 15:8 | | | | | CHINT11 | CHINT10 | CHINT9 | CHINT8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | BUSYCH | 7:0 | BUSYCH7 | BUSYCH6 | BUSYCH5 | BUSYCH4 | BUSYCH3 | BUSYCH2 | BUSYCH1 | BUSYCH0 |
| | | 15:8 | | | | | BUSYCH11 | BUSYCH10 | BUSYCH9 | BUSYCH8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | READYUSR | 7:0 | READYUSR7 | READYUSR6 | READYUSR5 | READYUSR4 | READYUSR3 | READYUSR2 | READYUSR1 | READYUSR0 |
| | | 15:8 | | | | | READYUSR11 | READYUSR10 | READYUSR9 | READYUSR8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | CHANNEL0 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | CHINTENCLR0 | 7:0 | | | | | | | EVD | OVR |
| 0x25 | CHINTENSET0 | 7:0 | | | | | | | EVD | OVR |
| 0x26 | CHINTFLAG0 | 7:0 | | | | | | | EVD | OVR |
| 0x27 | CHSTATUS0 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x28 | CHANNEL1 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x2C | CHINTENCLR1 | 7:0 | | | | | | | EVD | OVR |
| 0x2D | CHINTENSET1 | 7:0 | | | | | | | EVD | OVR |
| 0x2E | CHINTFLAG1 | 7:0 | | | | | | | EVD | OVR |
| 0x2F | CHSTATUS1 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x30 | CHANNEL2 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | CHINTENCLR2 | 7:0 | | | | | | | EVD | OVR |
| 0x35 | CHINTENSET2 | 7:0 | | | | | | | EVD | OVR |
| 0x36 | CHINTFLAG2 | 7:0 | | | | | | | EVD | OVR |
| 0x37 | CHSTATUS2 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x38 | CHANNEL3 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x3C | CHINTENCLR3 | 7:0 | | | | | | | EVD | OVR |
| 0x3D | CHINTENSET3 | 7:0 | | | | | | | EVD | OVR |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x3E | CHINTFLAG3 | 7:0 | | | | | | | EVD | OVR |
| 0x3F | CHSTATUS3 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x40 | CHANNEL4 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x44 | CHINTENCLR4 | 7:0 | | | | | | | EVD | OVR |
| 0x45 | CHINTENSET4 | 7:0 | | | | | | | EVD | OVR |
| 0x46 | CHINTFLAG4 | 7:0 | | | | | | | EVD | OVR |
| 0x47 | CHSTATUS4 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x48 | CHANNEL5 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C | CHINTENCLR5 | 7:0 | | | | | | | EVD | OVR |
| 0x4D | CHINTENSET5 | 7:0 | | | | | | | EVD | OVR |
| 0x4E | CHINTFLAG5 | 7:0 | | | | | | | EVD | OVR |
| 0x4F | CHSTATUS5 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x50 | CHANNEL6 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x54 | CHINTENCLR6 | 7:0 | | | | | | | EVD | OVR |
| 0x55 | CHINTENSET6 | 7:0 | | | | | | | EVD | OVR |
| 0x56 | CHINTFLAG6 | 7:0 | | | | | | | EVD | OVR |
| 0x57 | CHSTATUS6 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x58 | CHANNEL7 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x5C | CHINTENCLR7 | 7:0 | | | | | | | EVD | OVR |
| 0x5D | CHINTENSET7 | 7:0 | | | | | | | EVD | OVR |
| 0x5E | CHINTFLAG7 | 7:0 | | | | | | | EVD | OVR |
| 0x5F | CHSTATUS7 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x60 | CHANNEL8 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x64 | CHINTENCLR8 | 7:0 | | | | | | | EVD | OVR |
| 0x65 | CHINTENSET8 | 7:0 | | | | | | | EVD | OVR |
| 0x66 | CHINTFLAG8 | 7:0 | | | | | | | EVD | OVR |
| 0x67 | CHSTATUS8 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x68 | CHANNEL9 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x6C | CHINTENCLR9 | 7:0 | | | | | | | EVD | OVR |
| 0x6D | CHINTENSET9 | 7:0 | | | | | | | EVD | OVR |
| 0x6E | CHINTFLAG9 | 7:0 | | | | | | | EVD | OVR |
| 0x6F | CHSTATUS9 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x70 | CHANNEL10 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x74 | CHINTENCLR10 | 7:0 | | | | | | | EVD | OVR |
| 0x75 | CHINTENSET10 | 7:0 | | | | | | | EVD | OVR |
| 0x76 | CHINTFLAG10 | 7:0 | | | | | | | EVD | OVR |
| 0x77 | CHSTATUS10 | 7:0 | | | | | | | BUSYCH | RDYUSR |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x78 | CHANNEL11 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x7C | CHINTENCLR11 | 7:0 | | | | | | | EVD | OVR |
| 0x7D | CHINTENSET11 | 7:0 | | | | | | | EVD | OVR |
| 0x7E | CHINTFLAG11 | 7:0 | | | | | | | EVD | OVR |
| 0x7F | CHSTATUS11 | 7:0 | | | | | | | BUSYCH | RDYUSR |
| 0x80 | CHANNEL12 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x84 ... 0x87 | Reserved | | | | | | | | | |
| 0x88 | CHANNEL13 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x8C ... 0x8F | Reserved | | | | | | | | | |
| 0x90 | CHANNEL14 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x94 ... 0x97 | Reserved | | | | | | | | | |
| 0x98 | CHANNEL15 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x9C ... 0x9F | Reserved | | | | | | | | | |
| 0xA0 | CHANNEL16 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xA4 ... 0xA7 | Reserved | | | | | | | | | |
| 0xA8 | CHANNEL17 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xAC ... 0xAF | Reserved | | | | | | | | | |
| 0xB0 | CHANNEL18 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xB4 ... 0xB7 | Reserved | | | | | | | | | |
| 0xB8 | CHANNEL19 | 7:0 | | EVGEN[6:0] | | | | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xBC ... 0xBF | Reserved | | | | | | | | | |
| 0xC0 | CHANNEL20 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xC4 ... 0xC7 | Reserved | | | | | | | | | |
| 0xC8 | CHANNEL21 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xCC ... 0xCF | Reserved | | | | | | | | | |
| 0xD0 | CHANNEL22 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xD4 ... 0xD7 | Reserved | | | | | | | | | |
| 0xD8 | CHANNEL23 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xDC ... 0xDF | Reserved | | | | | | | | | |
| 0xE0 | CHANNEL24 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xE4 ... 0xE7 | Reserved | | | | | | | | | |
| 0xE8 | CHANNEL25 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xEC ... 0xEF | Reserved | | | | | | | | | |
| 0xF0 | CHANNEL26 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xF4 ... 0xF7 | Reserved | | | | | | | | | |
| 0xF8 | CHANNEL27 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xFC ... 0xFF | Reserved | | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0100 | CHANNEL28 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0104 ... 0x0107 | Reserved | | | | | | | | | |
| 0x0108 | CHANNEL29 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x010C ... 0x010F | Reserved | | | | | | | | | |
| 0x0110 | CHANNEL30 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0114 ... 0x0117 | Reserved | | | | | | | | | |
| 0x0118 | CHANNEL31 | 7:0 | | | | | EVGEN[6:0] | | | |
| | | 15:8 | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x011C ... 0x011F | Reserved | | | | | | | | | |
| 0x0120 | USER0 | 7:0 | | | CHANNEL[5:0] | | | | | |
| ... | | | | | | | | | | |
| 0x0162 | USER66 | 7:0 | | | CHANNEL[5:0] | | | | | |

## 31.7. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Refer to *Register Access Protection* and *PAC - Peripheral Access Controller*.

**Related Links**

PAC - Peripheral Access Controller

Register Access Protection


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SWRST |
| Access | | | | | | | | W |
| Reset | | | | | | | | 0 |

**Bit 0 – SWRST** Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the EVSYS to their initial state. It will always take precedence, meaning that all other writes in the same write-operation will be discarded.
**Note:** Before applying a Software Reset it is recommended to disable the event generators.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.2. Software Event

**Name:** SWEVT
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CHANNEL31 | CHANNEL30 | CHANNEL29 | CHANNEL28 | CHANNEL27 | CHANNEL26 | CHANNEL25 | CHANNEL24 |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CHANNEL23 | CHANNEL22 | CHANNEL21 | CHANNEL20 | CHANNEL19 | CHANNEL18 | CHANNEL17 | CHANNEL16 |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CHANNEL15 | CHANNEL14 | CHANNEL13 | CHANNEL12 | CHANNEL11 | CHANNEL10 | CHANNEL9 | CHANNEL8 |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CHANNEL7 | CHANNEL6 | CHANNEL5 | CHANNEL4 | CHANNEL3 | CHANNEL2 | CHANNEL1 | CHANNEL0 |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CHANNELx** Channel x Software Selection [x=0..7]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will trigger a software event for channel x.
These bits always return '0' when read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.3. Priority Control

**Name:** PRICTRL
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RREN | | | | PRI[3:0] | | | |
| Access | RW | | | | RW | RW | RW | RW |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 7 – RREN** Round-Robin Scheduling Enable
For details on scheduling schemes, refer to Interrupt Status and Interrupts Arbitration

| Value | Description |
|---|---|
| 0 | Static scheduling scheme for channels with level priority |
| 1 | Round-robin scheduling scheme for channels with level priority |

**Bits 3:0 – PRI[3:0]** Channel Priority Number
When round-robin arbitration is enabled (PRICTRL.RREN=1) for priority level, this register holds the channel number of the last EVSYS channel being granted access as the active channel with priority level. The value of this bit group is updated each time the INTPEND or any of CHINTFLAG registers are written.
When static arbitration is enabled (PRICTRL.RREN=0) for priority level, and the value of this bit group is nonzero, it will not affect the static priority scheme.
This bit group is not reset when round-robin scheduling gets disabled (PRICTRL.RREN written to zero).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.4. Channel Pending Interrupt

**Name:** INTPEND
**Offset:** 0x10
**Reset:** 0x4000

An interrupt that handles several channels should consult the INTPEND register to find out which channel number has priority (ignoring/filtering each channel that has its own interrupt line). An interrupt dedicated to only one channel must not use the INTPEND register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BUSY | READY | | | | | EVD | OVR |
| Access | R | R | | | | | RW | RW |
| Reset | 0 | 1 | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ID[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 15 – BUSY** Busy
This bit is read '1' when the event on a channel selected by Channel ID field (ID) has not been handled by all the event users connected to this channel.

**Bit 14 – READY** Ready
This bit is read '1' when all event users connected to the channel selected by Channel ID field (ID) are ready to handle incoming events on this channel.

**Bit 9 – EVD** Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.
When the event channel path is asynchronous, the EVD bit will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

**Bit 8 – OVR** Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVRx is '1'.
There are two possible overrun channel conditions:

- One or more of the event users on channel selected by Channel ID field (ID) are not ready when a new event occurs
- An event happens when the previous event on channel selected by Channel ID field (ID) has not yet been handled by all event users

When the event channel path is asynchronous, the OVR interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear it. It will also clear the corresponding flag in the Channel n Interrupt Flag Status and Clear register (CHINTFLAGn) of this peripheral, where n is determined by the Channel ID bit field (ID) in this register.

**Bits 3:0 – ID[3:0]** Channel ID
These bits store the channel number of the highest priority.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When the bits are written, indirect access to the corresponding Channel Interrupt Flag register is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.5. Interrupt Status

**Name:** INTSTATUS
**Offset:** 0x14
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CHINT11 | CHINT10 | CHINT9 | CHINT8 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CHINT7 | CHINT6 | CHINT5 | CHINT4 | CHINT3 | CHINT2 | CHINT1 | CHINT0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – CHINT** Channel x Pending Interrupt
This bit is set when Channel x has a pending interrupt.
This bit is cleared when the corresponding Channel x interrupts are disabled, or the source interrupt sources are cleared.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.6. Busy Channels

**Name:** BUSYCH
**Offset:** 0x18
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | BUSYCH11 | BUSYCH10 | BUSYCH9 | BUSYCH8 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BUSYCH7 | BUSYCH6 | BUSYCH5 | BUSYCH4 | BUSYCH3 | BUSYCH2 | BUSYCH1 | BUSYCH0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – BUSYCHx** Busy Channel x
This bit is set if an event occurs on channel x has not been handled by all event users connected to channel x.
This bit is cleared when channel x is idle.
When the event channel x path is asynchronous, this bit is always read '0'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.7. Ready Users

**Name:** READYUSR
**Offset:** 0x1C
**Reset:** 111111111111

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | READYUSR11 | READYUSR10 | READYUSR9 | READYUSR8 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | READYUSR7 | READYUSR6 | READYUSR5 | READYUSR4 | READYUSR3 | READYUSR2 | READYUSR1 | READYUSR0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – READYUSR** Ready User for Channel n
This bit is set when all event users connected to channel n are ready to handle incoming events on channel n.
This bit is cleared when at least one of the event users connected to the channel is not ready.
When the event channel n path is asynchronous, this bit is always read zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.8. Channel n Control

**Name:** CHANNELn
**Offset:** 0x20 + n*0x08 [n=0..31]
**Reset:** 0x00008000 for CHANNEL0-11, and 0x00000000 for CHANNEL12-31
**Property:** PAC Write-Protection

This register allows the user to configure channel n. To write to this register, do a single, 32-bit write of all the configuration data.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | | EDGSEL[1:0] | | PATH[1:0] | |
| Access | RW | RW | | | RW | RW | RW | RW |
| Reset | 1 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | EVGEN[6:0] | | | | | | |
| Access | | RW | RW | RW | RW | RW | RW | RW |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – ONDEMAND** Generic Clock On Demand
This bit is used to determine whether the generic clock is requested.
This bit has no effect for channels when asynchronous path is selected or for channels with asynchronous support only.
This bit is always read zero for channels with asynchronous support only.

| Value | Description |
|---|---|
| 0 | Generic clock for a channel is always on, if the channel is configured and generic clock source is enabled. |
| 1 | Generic clock is requested on demand while an event is handled |

**Bit 14 – RUNSTDBY** Run in Standby
This bit is used to define the behavior during standby sleep mode, for a resynchronized channel.
This bit has no effect for channels when asynchronous path is selected or for channels with asynchronous support only.
This bit is always read zero for channels with asynchronous support only.

| Value | Description |
|---|---|
| 0 | The channel is disabled in standby sleep mode. |
| 1 | The channel is not stopped in standby sleep mode and depends on the CHANNEL.ONDEMAND bit. |

**Bits 11:10 – EDGSEL[1:0]** Edge Detection Selection
These bits set the type of edge detection to be used on the channel.
These bits must be written to zero when using the asynchronous path.

| Value | Name | Description |
|---|---|---|
| 0x0 | NO_EVT_OUTPUT | No event output when using the resynchronized or synchronous path |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x1 | RISING_EDGE | Event detection only on the rising edge of the signal from the event generator |
| 0x2 | FALLING_EDGE | Event detection only on the falling edge of the signal from the event generator |
| 0x3 | BOTH_EDGES | Event detection on rising and falling edges of the signal from the event generator |

**Bits 9:8 – PATH[1:0]** Path Selection
These bits are used to choose which path will be used by the selected channel.
**Note:** The path choice can be limited by the channel source, see the table in USERm.

**Important:** Only EVSYS channel 0 to 11 can be configured as synchronous or resynchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | SYNCHRONOUS | Synchronous path |
| 0x1 | RESYNCHRONIZED | Resynchronized path |
| 0x2 | ASYNCHRONOUS | Asynchronous path |
| Other | - | Reserved |

**Bits 6:0 – EVGEN[6:0]** Event Generator Selection
These bits are used to choose the event generator to connect to the selected channel.

| Value | Name | Description |
|---|---|---|
| 0x00 | NONE | No event generator selected |
| 0x01 - 0x02 | OSCCTRL_XOSC_FAILx | XOSC fail detection x=0..1 |
| 0x03 | OSC32KCTRL_XOSC32K_FAIL | XOSC32K fail detection |
| 0x04 - 0x0B | RTC_PERx | RTC period x=0..7 |
| 0x0C - 0x0F | RTC_CMP | RTC comparison x=0..3 |
| 0x10 | RTC_TAMPER | RTC tamper detection |
| 0x11 | RTC_OVF | RTC overflow |
| 0x12 - 0x21 | EIC_EXTINT | EIC external interrupt x=0..15 |
| 0x22 - 0x25 | DMAC_CH | DMA channel x=0..3 |
| 0x26 | PAC_ACCERR | PAC Acc. error |
| 0x27 | Reserved | - |
| 0x28 | Reserved | - |
| 0x29 | TCC0_OVF | TCC0 Overflow |
| 0x2A | TCC0_TRG | TCC0 Trigger Event |
| 0x2B | TCC0_CNT | TCC0 Counter |
| 0x2C - 0x31 | TCC0_MCx | TCC0 Match/Compare x=0..5 |
| 0x32 | TCC1_OVF | TCC1 Overflow |
| 0x33 | TCC1_TRG | TCC1 Trigger Event |
| 0x34 | TCC1_CNT | TCC1 Counter |
| 0x35 - 0x38 | TCC1_MCx | TCC1 Match/Compare x=0..3 |
| 0x39 | TCC2_OVF | TCC2 Overflow |
| 0x3A | TCC2_TRG | TCC2 Trigger Event |
| 0x3B | TCC2_CNT | TCC2 Counter |
| 0x3C - 0x3E | TCC2_MCx | TCC2 Match/Compare x=0..2 |
| 0x3F | TCC3_OVF | TCC3 Overflow |
| 0x40 | TCC3_TRG | TCC3 Trigger Event |
| 0x41 | TCC3_CNT | TCC3 Counter |
| 0x42 - 0x43 | TCC3_MCx | TCC3 Match/Compare x=0..1 |
| 0x44 | TCC4_OVF | TCC4 Overflow |
| 0x45 | TCC4_TRG | TCC4 Trigger Event |
| 0x46 | TCC4_CNT | TCC4 Counter |
| 0x47 - 0x48 | TCC4_MCx | TCC4 Match/Compare x=0..1 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| **EVGEN** (continued) | | |
|---|---|---|
| **Value** | **Name** | **Description** |
| 0x49 | TC0_OVF | TC0 Overflow |
| 0x4A - 0x4B | TC0_MCx | TC0 Match/Compare x=0..1 |
| 0x4C | TC1_OVF | TC1 Overflow |
| 0x4D - 0x4E | TC1_MCx | TC1 Match/Compare x=0..1 |
| 0x4F | TC2_OVF | TC2 Overflow |
| 0x50 - 0x51 | TC2_MCx | TC2 Match/Compare x=0..1 |
| 0x52 | TC3_OVF | TC3 Overflow |
| 0x53 - 0x54 | TC3_MCx | TC3 Match/Compare x=0..1 |
| 0x55 | TC4_OVF | TC4 Overflow |
| 0x56 - 0x57 | TC4_MCx | TC4 Match/Compare x=0..1 |
| 0x58 | TC5_OVF | TC5 Overflow |
| 0x59 - 0x5A | TC5_MCx | TC5 Match/Compare x=0..1 |
| 0x5B | TC6_OVF | TC6 Overflow |
| 0x5C - 0x5D | TC6_MCx | TC6 Match/Compare x=0..1 |
| 0x5E | TC7_OVF | TC7 Overflow |
| 0x5F - 0x60 | TC7_MCx | TC7 Match/Compare x=0..1 |
| 0x61 | PDEC_OVF | PDEC Overflow |
| 0x62 | PDEC_ERR | PDEC Error |
| 0x63 | PDEC_DIR | PDEC Direction |
| 0x64 | PDEC_VLC | PDEC VLC |
| 0x65 - 0x66 | PDEC_MCx | PDEC MCx x=0..1 |
| 0x67 | ADC0_RESRDY | ADC0 RESRDY |
| 0x68 | ADC0_WINMON | ADC0 Window Monitor |
| 0x69 | ADC1_RESRDY | ADC1 RESRDY |
| 0x6A | ADC1_WINMON | ADC1 Window Monitor |
| 0x6B - 0x6C | AC_COMPx | AC Comparator, x=0..1 |
| 0x6D | AC_WIN | AC0 Window |
| 0x6E - 0x6F | DAC_EMPTYx | DAC empty, x=0..1 |
| 0x70 - 0x71 | DAC_RESRDYx | DAC RSRDY, x=0..1 |
| 0x72 | GMAC_TSU_CMP | GMAC Timestamp CMP |
| 0x73 | TRNG_READY | TRNG ready |
| 0x74 - 0x77 | CCL_LUTOUT | CCL LUTOUT |
| 0x78-0x7F | Reserved | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 31.7.9. Channel n Interrupt Enable Clear

**Name:** CHINTENCLR
**Offset:** 0x24 + n*0x08 [n=0..11]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EVD | OVR |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – EVD** Channel Event Detected Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Event Detected Channel Interrupt Enable bit, which disables the Event Detected Channel interrupt.

| Value | Description |
|---|---|
| 0 | The Event Detected Channel interrupt is disabled. |
| 1 | The Event Detected Channel interrupt is enabled. |

**Bit 0 – OVR** Channel Overrun Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Channel Interrupt Enable bit, which disables the Overrun Channel interrupt.

| Value | Description |
|---|---|
| 0 | The Overrun Channel interrupt is disabled. |
| 1 | The Overrun Channel interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.10. Channel n Interrupt Enable Set

**Name:** CHINTENSET
**Offset:** 0x25 + n*0x08 [n=0..11]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EVD | OVR |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – EVD** Channel Event Detected Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Event Detected Channel Interrupt Enable bit, which enables the Event Detected Channel interrupt.

| Value | Description |
|---|---|
| 0 | The Event Detected Channel interrupt is disabled. |
| 1 | The Event Detected Channel interrupt is enabled. |

**Bit 0 – OVR** Channel Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Channel Interrupt Enable bit, which enables the Overrun Channel interrupt.

| Value | Description |
|---|---|
| 0 | The Overrun Channel interrupt is disabled. |
| 1 | The Overrun Channel interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.11. Channel n Interrupt Flag Status and Clear

**Name:** CHINTFLAG
**Offset:** 0x26 + n*0x08 [n=0..11]
**Reset:** 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EVD | OVR |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – EVD** Channel Event Detected
This flag is set on the next CLK_EVSYS_APB cycle when an event is being propagated through the channel, and an interrupt request will be generated if CHINTENCLR/SET.EVD is '1'.
When the event channel path is asynchronous, the EVD interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Event Detected Channel interrupt flag.

**Bit 0 – OVR** Channel Overrun
This flag is set on the next CLK_EVSYS cycle after an overrun channel condition occurs, and an interrupt request will be generated if CHINTENCLR/SET.OVRx is '1'.
There are two possible overrun channel conditions:

- One or more of the event users on the channel are not ready when a new event occurs.
- An event happens when the previous event on channel has not yet been handled by all event users.

When the event channel path is asynchronous, the OVR interrupt flag will not be set.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Channel interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.12. Channel n Status

**Name:** CHSTATUSn
**Offset:** 0x27 + n*0x08 [n=0..11]
**Reset:** 0x01

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BUSYCH | RDYUSR |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – BUSYCH** Busy Channel
This bit is cleared when channel is idle.
This bit is set if an event on channel has not been handled by all event users connected to channel.
When the event channel path is asynchronous, this bit is always read '0'.

**Bit 0 – RDYUSR** Ready User
This bit is cleared when at least one of the event users connected to the channel is not ready.
This bit is set when all event users connected to channel are ready to handle incoming events on the channel.
When the event channel path is asynchronous, this bit is always read zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 31.7.13. Event User m

**Name:** USERm
**Offset:** 0x0120 + m*0x01 [m=0..66]
**Reset:** 0x0
**Property:** PAC Write-Protection

**Table 31-2.** User Multiplexer Number m

| USER*m* | User Multiplexer | Description | Path Type(1) |
|---|---|---|---|
| m = 0 | RTC_TAMPER | RTC Tamper | A |
| m = 1..4 | PORT_EV0..3 | PORT Event 0..3 | A |
| m = 5..12 | DMAC_CH0..7 | Channel 0..7 | S, R |
| m = 13 | - | Reserved | - |
| m = 14 | CM4_TRACE_START | CM4 trace start | S, R |
| m = 15 | CM4_TRACE_STOP | CM4 trace stop | S, R |
| m = 16 | CM4_TRACE_TRIG | CM4 trace trigger | S, R |
| m = 17..18 | TCC0 EV0..1 | TCC0 EVx | A, S, R |
| m = 19..24 | TCC0 MC0..5 | TCC0 MCx | A, S, R |
| m = 25..26 | TCC1 EV0..1 | TCC1 EVx | A, S, R |
| m = 27..30 | TCC1 MC0..3 | TCC1 MCx | A, S, R |
| m = 31..32 | TCC2 EV0..1 | TCC2 EVx | A, S, R |
| m = 33..35 | TCC2 MC0..2 | TCC2 MCx | A, S, R |
| m = 36..37 | TCC3 EV0..1 | TCC3 EVx | A, S, R |
| m = 38..39 | TCC3 MC0..1 | TCC3 MCx | A, S, R |
| m = 40..41 | TCC4 EV0..1 | TCC4 EVx | A, S, R |
| m = 42..43 | TCC4 MC0..1 | TCC4 MCx | A, S, R |
| m = 44..51 | TC0..7 EVU | TC0..7 EVU | A, S, R |
| m = 52..54 | PDEC_EVU 0..2 | PDEC EVU x | A, S, R |
| m = 55 | ADC0 START | ADC0 start conversion | A, S, R |
| m = 56 | ADC0 SYNC | Flush ADC0 | A, S, R |
| m = 57 | ADC1 START | ADC1 start conversion | A, S, R |
| m = 58 | ADC1 SYNC | Flush ADC1 | A, S, R |
| m = 59..60 | AC_SOC 0..1 | AC SOC x | A |
| m = 61..62 | DAC_START0..1 | DAC0..1 start conversion | A |
| m = 63..66 | CCL_LUTIN 0..3 | CCL input | A |
| others | Reserved | - | - |

**Note:**

1. A = Asynchronous path, S = Synchronous path, R = Resynchronized path.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | CHANNEL[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:0 – CHANNEL[5:0]** Channel Event Selection
These bits select channel n to connect to the event user m.
**Note:** A value x of this bit field selects channel n = x-1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0x00 | No channel selected |
| 0x01 | Channel 0 selected |
| 0x02 | Channel 1 selected |
| 0x03 | Channel 2 selected |
| 0x04 | Channel 3 selected |
| 0x05 | Channel 4 selected |
| 0x06 | Channel 5 selected |
| 0x07 | Channel 6 selected |
| 0x08 | Channel 7 selected |
| 0x09 | Channel 8 selected |
| 0x0A | Channel 9 selected |
| 0x0B | Channel 10 selected |
| 0x0C | Channel 11 selected |
| 0x0D | Channel 12 selected |
| 0x0E | Channel 13 selected |
| 0x0F | Channel 14 selected |
| 0x10 | Channel 15 selected |
| 0x11 | Channel 16 selected |
| 0x12 | Channel 17 selected |
| 0x13 | Channel 18 selected |
| 0x14 | Channel 19 selected |
| 0x15 | Channel 20 selected |
| 0x16 | Channel 21 selected |
| 0x17 | Channel 22 selected |
| 0x18 | Channel 23 selected |
| 0x19 | Channel 24 selected |
| 0x1A | Channel 25 selected |
| 0x1B | Channel 26 selected |
| 0x1C | Channel 27 selected |
| 0x1D | Channel 28 selected |
| 0x1E | Channel 29 selected |
| 0x1F | Channel 30 selected |
| 0x20 | Channel 31 selected |
| other | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 32. PORT - I/O Pin Controller

## 32.1. Overview

The I/O Pin Controller (PORT) controls the I/O pins of the device. The I/O pins are organized in a series of groups, collectively referred to as a PORT group. Each PORT group can have up to 32 pins that can be configured and controlled individually or as a group. The number of PORT groups on a device may depend on the package or number of pins. Each pin may either be used for general purpose I/O under direct application control or be assigned to an embedded device peripheral. When used for general purpose I/O, each pin can be configured as input or output, with a highly configurable driver and pull settings.

All I/O pins have true read-modify-write functionality when used for general purpose I/O. The direction or the output value of one or more pins may be changed (set, Reset or toggled) explicitly without unintentionally changing the state of any other pins in the same port group by a single, atomic 8-, 16- or 32-bit write.

The PORT is connected to the high-speed bus matrix through an AHB/APB bridge.

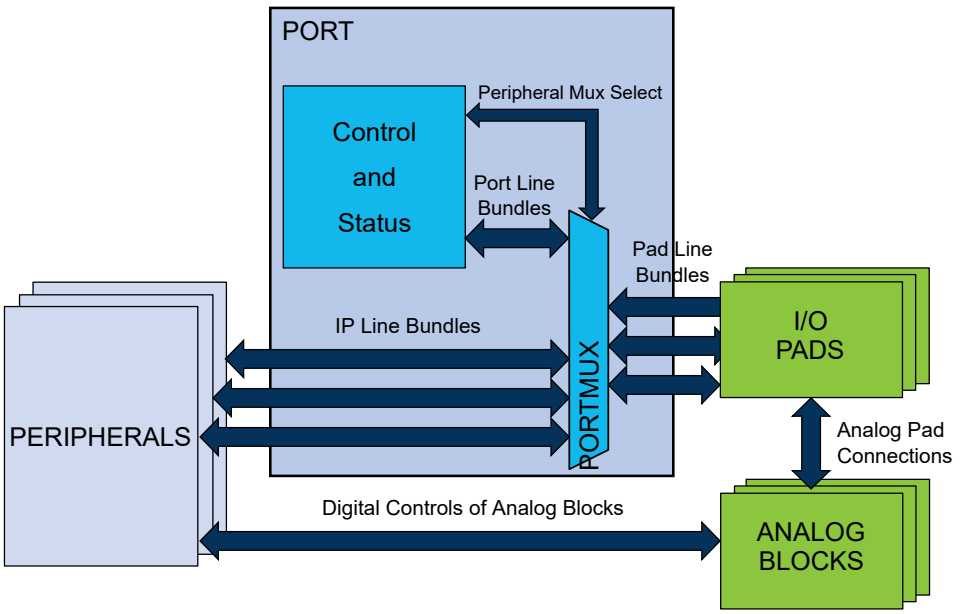
## 32.2. Features

- Selectable Input and Output Configuration for Each Individual Pin
- Software-controlled Multiplexing of Peripheral Functions on I/O Pins
- Flexible Pin Configuration Through a Dedicated Pin Configuration Register
- Configurable Output Driver and Pull Settings:
  - Totem-pole (push-pull)
  - Pull configuration
  - Driver strength
- Configurable Input Buffer and Pull Settings:
  - Internal pull-up or pull-down
  - Input sampling criteria
  - Input buffer can be disabled if not needed for lower power consumption
  - Read-Modify-Write support for output value (OUTCLR/OUTSET/OUTGL) and pin direction (DIRCLR/DIRSET/DIRTGL)
- Input Event:
  - Up to four input event pins for each PORT group
  - SET/CLEAR/TOGGLE event actions for each event input on output value of a pin
  - Can be output to pin


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 32.3. Block Diagram

**Figure 32-1.** PORT Block Diagram




## 32.4. Signal Description

**Table 32-1.** Signal Description for PORT

| Signal name | Type | Description |
|---|---|---|
| Pxy | Digital I/O | General purpose I/O pin y in group x |

Refer to the *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

**Related Links**
I/O Multiplexing and Considerations

## 32.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly as follows.

### 32.5.1. I/O Lines

The I/O lines of the PORT are mapped to pins of the physical device. The following naming scheme is used:

Each line bundle with up to 32 lines is assigned an identifier 'xy', with letter x=A, B, C... and two-digit number y=00, 01, ...31. Examples: A24, C03.

PORT pins are labeled 'Pxy' accordingly, for example PA24, PC03. This identifies each pin in the device uniquely.

Each pin may be controlled by one or more peripheral multiplexer settings, which allows the pad to be routed internally to a dedicated peripheral function. When the setting is enabled, the selected peripheral has control over the Output state of the pad, as well as the ability to read the current Physical Pad state. Refer to *I/O Multiplexing and Considerations* for details.

Device-specific configurations may cause some lines (and the corresponding Pxy pin) not to be implemented.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
I/O Multiplexing and Considerations

### 32.5.2. Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled.

When the device is set to the BACKUP sleep mode, even if the PORT configuration registers and input synchronizers will lose their contents (these will not be restored when PORT is powered up again), the latches in the pads will keep their current configuration, such as the output value and pull settings. Refer to the Power Manager documentation for more features related to the I/O lines configuration in and out of BACKUP mode.

The PORT peripheral will continue operating in any Sleep mode where its source clock is running.

**Related Links**
I/O Lines Retention in HIBERNATE or BACKUP Mode

### 32.5.3. Clocks

The PORT bus clock (CLK_PORT_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_PORT_APB can be found in the *Peripheral Clock Masking* section in *MCLK – Main Clock*.

The PORT requires an APB clock, which may be divided from the CPU main clock and allows the CPU to access the registers of PORT through the high-speed matrix and the AHB/APB bridge.

One clock cycle latency can be observed on the APB access in case of concurrent PORT accesses.

**Related Links**
MCLK – Main Clock

### 32.5.4. DMA

Not applicable.

### 32.5.5. Interrupts

Not applicable.

### 32.5.6. Events

The events of this peripheral are connected to the Event System.

The output of an event to a pin through PORT is always asynchronous. This must be configured in the Event System by writing ASYNCHRONOUS to the Path Selection bits in the respective Channel n Control register of the Event System (EVSYS.CHANNELn.PATH).

**Related Links**
Event System (EVSYS)

### 32.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation.

### 32.5.8. Register Access Protection

All registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC).

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

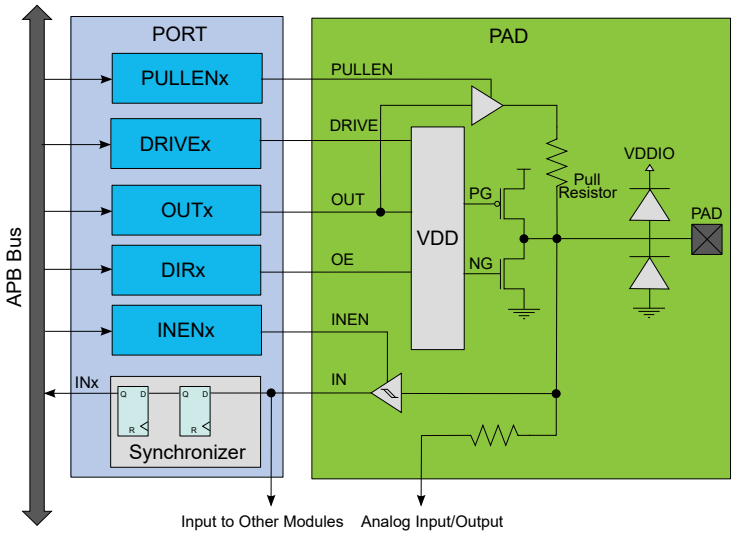
**Related Links**
PAC - Peripheral Access Controller

### 32.5.9. Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. Selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

## 32.6. Functional Description

**Figure 32-2.** Overview of the PORT

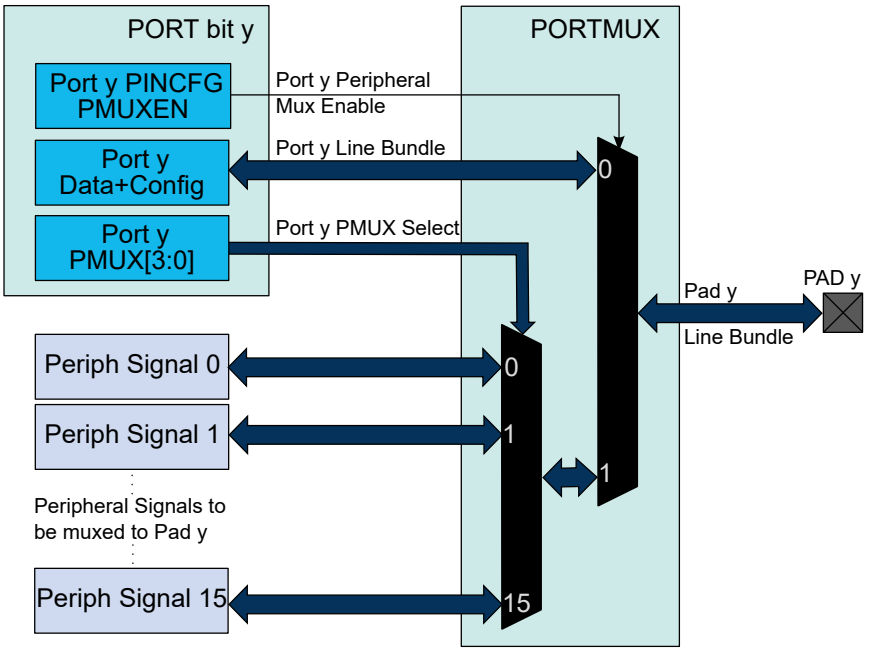



### 32.6.1. Principle of Operation

Each PORT group of up to 32 pins is controlled by the registers in PORT, as described in the figure. These registers in PORT are duplicated for each PORT group, with increasing base addresses using an offset of 0x80 between groups. The number of PORT groups may depend on the package/ number of pins.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 32-3.** Overview of the peripheral functions multiplexing




The I/O pins of the device are controlled by PORT peripheral registers. Each port pin has a corresponding bit in the Data Direction (DIR) and Data Output Value (OUT) registers to enable that pin as an output and to define the Output state.

The direction of each pin in a PORT group is configured by the DIR register. If a bit in DIR is set to '1', the corresponding pin is configured as an output pin. If a bit in DIR is set to '0', the corresponding pin is configured as an input pin.

When the direction is set as output, the corresponding bit in the OUT register will set the level of the pin. If bit y in OUT is written to '1', pin y is driven HIGH. If bit y in OUT is written to '0', pin y is driven LOW. Pin configuration can be set by Pin Configuration (PINCFGy) registers, with y=00, 01, ..31 representing the pin number within the group.

The Data Input Value (IN) is set as the input value of a port pin with resynchronization to the PORT clock. To reduce power consumption, these input synchronizers can be clocked only when system requires reading the input value, as specified in the SAMPLING field of the Control register (CTRL). The value of the pin can always be read, whether the pin is configured as input or output. If the Input Enable bit in the Pin Configuration registers (PINCFGy.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGy register (PINCFGy.PMUXEN) can be written to '1' to enable the connection between peripheral functions and individual I/O pins. The Peripheral Multiplexing n (PMUXn) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, and connect the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

## 32.6.2. Basic Operation

### 32.6.2.1. Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled, even if there is no clock running.

However, specific pins, such as those used for connection to a debugger, may be configured differently, as required by their special function.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 32.6.2.2. Operation

Each I/O pin Pxy can be controlled by the registers in PORT. Each PORT group x has its own set of PORT registers, with a base address at byte address (PORT + 0x80 * group index) (A corresponds to group index 0, B to 1, etc...). Within that set of registers, the pin index is y, from 0 to 31.

Refer to *I/O Multiplexing and Considerations* for details on available pin configuration and PORT groups.

**Configuring Pins as Output**

To use pin Pxy as an *output*, write bit y of the DIR register to '1'. This can also be done by writing bit y in the DIRSET register to '1' - this will avoid disturbing the configuration of other pins in that group. The y bit in the OUT register must be written to the desired output value.

Similarly, writing an OUTSET bit to '1' will set the corresponding bit in the OUT register to '1'. Writing a bit in OUTCLR to '1' will set that bit in OUT to zero. Writing a bit in OUTTGL to '1' will toggle that bit in OUT.

**Configuring Pins as Input**

To use pin Pxy as an *input*, bit y in the DIR register must be written to '0'. This can also be done by writing bit y in the DIRCLR register to '1' - this will avoid disturbing the configuration of other pins in that group. The input value can be read from bit y in register IN as soon as the INEN bit in the Pin Configuration register (PINCFGy.INEN) is written to '1' to enable the pin's input buffer.

By default, the input synchronizer is clocked only when an input read is requested. This will delay the read operation by two cycles of the PORT clock. To remove the delay, the input synchronizers for each PORT group of eight pins can be configured to be always active, but this will increase power consumption. This is enabled by writing '1' to the corresponding SAMPLINGn bit field of the CTRL register, see CTRL.SAMPLING for details.

**Using Alternative Peripheral Functions**

To use pin Pxy as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGy register must be '1'. The PINCFGy register for pin Pxy is at byte offset (PINCFG0 + y).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXn register. The PMUXO/PMUXE is at byte offset PMUX0 + (y/2). The chosen peripheral must also be configured and enabled.

**Related Links**

I/O Multiplexing and Considerations

### 32.6.3. I/O Pin Configuration

The Pin Configuration register (PINCFGy) is used for additional I/O pin configuration. A pin can be set in a totem-pole or pull configuration.

As pull configuration is done through the Pin Configuration register, all intermediate PORT states during switching of pin direction and pin values are avoided.

The I/O pin configurations are described further in this chapter, and summarized in Table 32-2.

#### 32.6.3.1. Pin Configurations Summary

**Table 32-2.** Pin Configurations Summary

| DIR | INEN | PULLEN | OUT | Configuration |
|---|---|---|---|---|
| 0 | 0 | 0 | X | Reset or analog I/O: all digital disabled |
| 0 | 0 | 1 | 0 | Pull-down; input disabled |
| 0 | 0 | 1 | 1 | Pull-up; input disabled |
| 0 | 1 | 0 | X | Input |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 32-2.** Pin Configurations Summary (continued)

| DIR | INEN | PULLEN | OUT | Configuration |
|---|---|---|---|---|
| 0 | 1 | 1 | 0 | Input with pull-down |
| 0 | 1 | 1 | 1 | Input with pull-up |
| 1 | 0 | X | X | Output; input disabled |
| 1 | 1 | X | X | Output; input enabled |

### 32.6.3.2. Input Configuration

**Figure 32-4.** I/O configuration - Standard Input

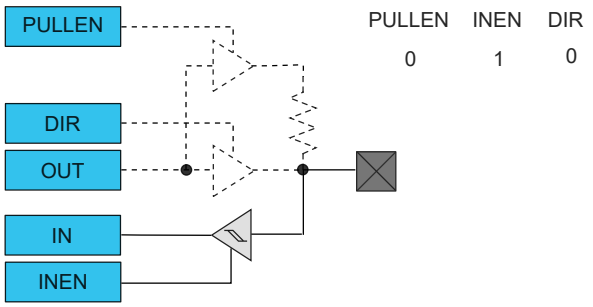
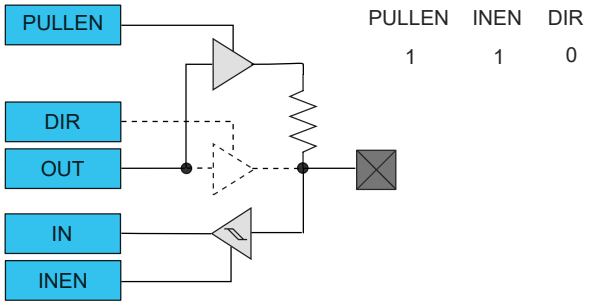



**Figure 32-5.** I/O Configuration - Input with Pull



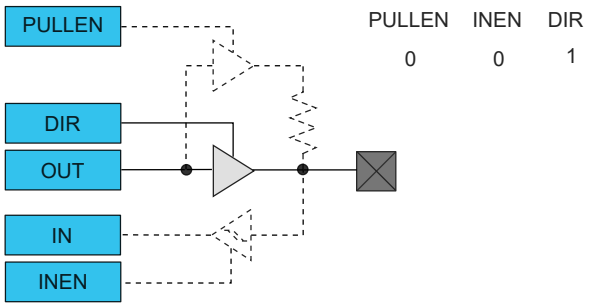

**Note:** When pull is enabled, the pull value is defined by the OUT value.

### 32.6.3.3. Totem-Pole Output

When configured for totem-pole (push-pull) output, the pin is driven low or high according to the corresponding bit setting in the OUT register. In this configuration there is no current limitation for sink or source other than what the pin is capable of. If the pin is configured for input, the pin will float if no external pull is connected.

**Note:** Enabling the output driver will automatically disable pull.

**Figure 32-6.** I/O Configuration - Totem-Pole Output with Disabled Input





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 32-7.** I/O Configuration - Totem-Pole Output with Enabled Input

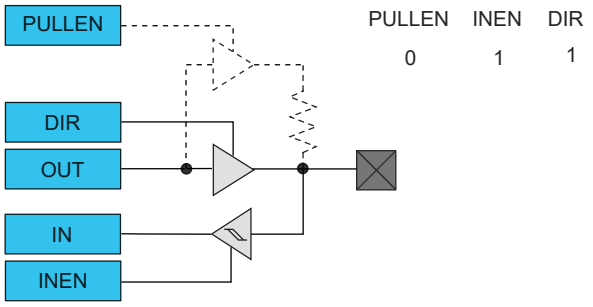
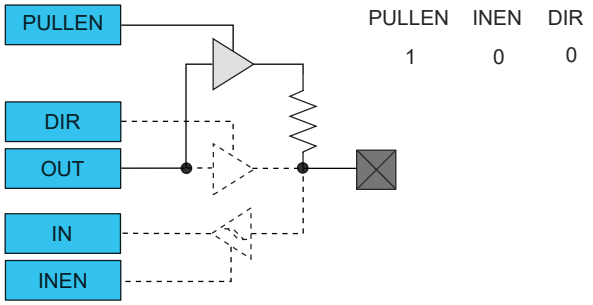


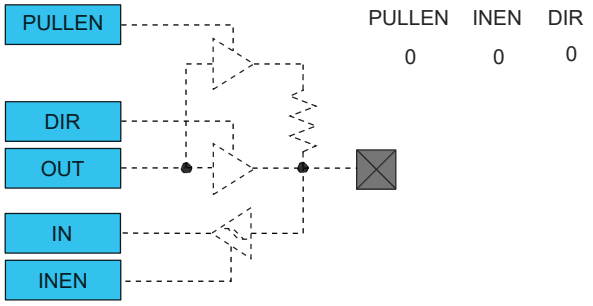

**Figure 32-8.** I/O Configuration - Output with Pull




### 32.6.3.4. Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

**Figure 32-9.** I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled




## 32.6.4. Events

The PORT allows input events to control individual I/O pins. These input events (EVU0-3) are generated by the EVSYS module and can originate from a different clock domain than the PORT module's clock domain.

The PORT can perform the following actions:

- **Output (OUT):** I/O pin will be set when the incoming event has a high level ('1'), and cleared when the incoming event has a low-level ('0')
- **Set (SET):** I/O pin will be set when an incoming event is detected
- **Clear (CLR):** I/O pin will be cleared when an incoming event is detected
- **Toggle (TGL):** I/O pin will toggle when an incoming event is detected

The Output event is sent to the pin without any internal latency. For SET, CLEAR, and TOGGLE event actions, the action will be executed up to three clock cycles after a rising edge.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** In Standby mode, only the Out action is possible, and the Set, Clear, and Toggle actions are not available.

The event actions can be configured with the Event Action m bit group in the Event Input Control register( EVCTRL.EVACTn). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIn) enables the corresponding action on input event. Writing '0' to this bit disables the corresponding action on input event. Several actions can be enabled for incoming events. If several events are connected to the peripheral, any enabled action will be taken for any of the incoming events. Refer to the section " EVSYS – Event System" for additional information on configuring the Event System.

Each event input can address one and only one I/O pin at a time. The selection of the pin is indicated by the PORT Event Pin Identifier of the Event Input Control register (EVCTR.PIDn). On the other hand, one I/O pin can be addressed by up to four different input events. To avoid action conflict on the output value of the register (OUT) of this particular I/O pin, only one action is performed according to the table below. This truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

**Table 32-3.** Priority on Simultaneous SET/CLR/TGL Event Actions

| EVACT0 | EVACT1 | EVACT2 | EVACT3 | Executed Event Action |
|---|---|---|---|---|
| SET | SET | SET | SET | SET |
| CLR | CLR | CLR | CLR | CLR |
| All Other Combinations | | | | TGL |

Be careful when the event is Output (OUT) to pin. Because the events are received asynchronously, the I/O pin may have unpredictable levels, depending on the timing of when the events are received. When several events are output to the same pin, the lowest event line will get the access. All other events will be ignored.

**Related Links**
Event System (EVSYS)

### 32.6.5. PORT Access Priority

The PORT is accessed by different systems:

- The ARM® CPU through the high-speed matrix and the AHB/APB bridge (APB)
- EVSYS through four asynchronous input events

The following priority is adopted:

1. APB
2. EVSYS input events, except for events with EVCTRL.EVACTn=OUT, where the output pin directly follows the event input signal, independently of the OUT register value.

For input events that require different actions on the same I/O pin, refer to Events.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 32.7. Register Summary

The I/O pins are assembled in pin groups with up to 32 pins. Group 0 consists of the PA pins, and group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | DIR | 7:0 | DIR[7:0] | | | | | | | |
| | | 15:8 | DIR[15:8] | | | | | | | |
| | | 23:16 | DIR[23:16] | | | | | | | |
| | | 31:24 | DIR[31:24] | | | | | | | |
| 0x04 | DIRCLR | 7:0 | DIRCLR[7:0] | | | | | | | |
| | | 15:8 | DIRCLR[15:8] | | | | | | | |
| | | 23:16 | DIRCLR[23:16] | | | | | | | |
| | | 31:24 | DIRCLR[31:24] | | | | | | | |
| 0x08 | DIRSET | 7:0 | DIRSET[7:0] | | | | | | | |
| | | 15:8 | DIRSET[15:8] | | | | | | | |
| | | 23:16 | DIRSET[23:16] | | | | | | | |
| | | 31:24 | DIRSET[31:24] | | | | | | | |
| 0x0C | DIRTGL | 7:0 | DIRTGL[7:0] | | | | | | | |
| | | 15:8 | DIRTGL[15:8] | | | | | | | |
| | | 23:16 | DIRTGL[23:16] | | | | | | | |
| | | 31:24 | DIRTGL[31:24] | | | | | | | |
| 0x10 | OUT | 7:0 | OUT[7:0] | | | | | | | |
| | | 15:8 | OUT[15:8] | | | | | | | |
| | | 23:16 | OUT[23:16] | | | | | | | |
| | | 31:24 | OUT[31:24] | | | | | | | |
| 0x14 | OUTCLR | 7:0 | OUTCLR[7:0] | | | | | | | |
| | | 15:8 | OUTCLR[15:8] | | | | | | | |
| | | 23:16 | OUTCLR[23:16] | | | | | | | |
| | | 31:24 | OUTCLR[31:24] | | | | | | | |
| 0x18 | OUTSET | 7:0 | OUTSET[7:0] | | | | | | | |
| | | 15:8 | OUTSET[15:8] | | | | | | | |
| | | 23:16 | OUTSET[23:16] | | | | | | | |
| | | 31:24 | OUTSET[31:24] | | | | | | | |
| 0x1C | OUTTGL | 7:0 | OUTTGL[7:0] | | | | | | | |
| | | 15:8 | OUTTGL[15:8] | | | | | | | |
| | | 23:16 | OUTTGL[23:16] | | | | | | | |
| | | 31:24 | OUTTGL[31:24] | | | | | | | |
| 0x20 | IN | 7:0 | IN[7:0] | | | | | | | |
| | | 15:8 | IN[15:8] | | | | | | | |
| | | 23:16 | IN[23:16] | | | | | | | |
| | | 31:24 | IN[31:24] | | | | | | | |
| 0x24 | CTRL | 7:0 | SAMPLING[7:0] | | | | | | | |
| | | 15:8 | SAMPLING[15:8] | | | | | | | |
| | | 23:16 | SAMPLING[23:16] | | | | | | | |
| | | 31:24 | SAMPLING[31:24] | | | | | | | |
| 0x28 | WRCONFIG | 7:0 | PINMASK[7:0] | | | | | | | |
| | | 15:8 | PINMASK[15:8] | | | | | | | |
| | | 23:16 | | DRVSTR | | | | PULLEN | INEN | PMUXEN |
| | | 31:24 | HWSEL | WRPINCFG | | WRPMUX | | PMUX[3:0] | | |
| 0x2C | EVCTRL | 7:0 | PORTEI0 | EVACT0[1:0] | | PID0[4:0] | | | | |
| | | 15:8 | PORTEI1 | EVACT1[1:0] | | PID1[4:0] | | | | |
| | | 23:16 | PORTEI2 | EVACT2[1:0] | | PID2[4:0] | | | | |
| | | 31:24 | PORTEI3 | EVACT3[1:0] | | PID3[4:0] | | | | |
| 0x30 | PMUX0 | 7:0 | PMUXO[3:0] | | | | PMUXE[3:0] | | | |
| ... | | | | | | | | | | |
| 0x3F | PMUX15 | 7:0 | PMUXO[3:0] | | | | PMUXE[3:0] | | | |
| 0x40 | PINCFG0 | 7:0 | | DRVSTR | | | | PULLEN | INEN | PMUXEN |
| ... | | | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x5F | PINCFG31 | 7:0 | | DRVSTR | | | | PULLEN | INEN | PMUXEN |

## 32.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.1. Data Direction

**Name:** DIR
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to configure one or more I/O pins as an input or output. This register can be manipulated without doing a read-modify-write operation by using the Data Direction Toggle (DIRTGL), Data Direction Clear (DIRCLR) and Data Direction Set (DIRSET) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIR[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DIR[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DIR[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIR[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DIR[31:0]** Port Data Direction
These bits set the data direction for the individual I/O pins in the PORT group.

| Value | Description |
|---|---|
| 0 | The corresponding I/O pin in the PORT group is configured as an input. |
| 1 | The corresponding I/O pin in the PORT group is configured as an output. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.2. Data Direction Clear

**Name:** DIRCLR
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to set one or more I/O pins as an input, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Set (DIRSET) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIRCLR[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DIRCLR[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DIRCLR[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIRCLR[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DIRCLR[31:0]** Port Data Direction Clear
Writing a '0' to a bit has no effect.
Writing a '1' to a bit will clear the corresponding bit in the DIR register, which configures the I/O pin as an input.

| Value | Description |
|---|---|
| 0 | The corresponding I/O pin in the PORT group will keep its configuration. |
| 1 | The corresponding I/O pin in the PORT group is configured as input. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.3. Data Direction Set

**Name:** DIRSET
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to set one or more I/O pins as an output, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Toggle (DIRTGL) and Data Direction Clear (DIRCLR) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIRSET[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DIRSET[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DIRSET[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIRSET[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DIRSET[31:0]** Port Data Direction Set
Writing '0' to a bit has no effect.
Writing '1' to a bit will set the corresponding bit in the DIR register, which configures the I/O pin as an output.

| Value | Description |
|---|---|
| 0 | The corresponding I/O pin in the PORT group will keep its configuration. |
| 1 | The corresponding I/O pin in the PORT group is configured as an output. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.4. Data Direction Toggle

**Name:** DIRTGL
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to toggle the direction of one or more I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Direction (DIR), Data Direction Set (DIRSET) and Data Direction Clear (DIRCLR) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIRTGL[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DIRTGL[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DIRTGL[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIRTGL[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DIRTGL[31:0]** Port Data Direction Toggle
Writing '0' to a bit has no effect.
Writing '1' to a bit will toggle the corresponding bit in the DIR register, which reverses the direction of the I/O pin.

| Value | Description |
|---|---|
| 0 | The corresponding I/O pin in the PORT group will keep its configuration. |
| 1 | The direction of the corresponding I/O pin is toggled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.5. Data Output Value

**Name:** OUT
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register sets the data output drive value for the individual I/O pins in the PORT.

This register can be manipulated without doing a read-modify-write operation by using the Data Output Value Clear (OUTCLR), Data Output Value Set (OUTSET), and Data Output Value Toggle (OUTTGL) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | OUT[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | OUT[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OUT[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OUT[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – OUT[31:0]** PORT Data Output Value
For pins configured as outputs via the Data Direction register (DIR), these bits set the logical output drive level.
For pins configured as inputs via the Data Direction register (DIR) and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN), these bits will set the input pull direction.

| Value | Description |
|---|---|
| 0 | The I/O pin output is driven low, or the input is connected to an internal pull-down. |
| 1 | The I/O pin output is driven high, or the input is connected to an internal pull-up. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.6. Data Output Value Clear

**Name:** OUTCLR
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to set one or more output I/O pin drive levels low, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Set (OUTSET) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | OUTCLR[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | OUTCLR[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OUTCLR[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OUTCLR[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – OUTCLR[31:0]** PORT Data Output Value Clear
Writing '0' to a bit has no effect.
Writing '1' to a bit will clear the corresponding bit in the OUT register. Pins configured as outputs via the Data Direction register (DIR) will be set to low output drive level. Pins configured as inputs via DIR and with pull enabled via the Pull Enable bit in the Pin Configuration register (PINCFG.PULLEN) will set the input pull direction to an internal pull-down.

| Value | Description |
|---|---|
| 0 | The corresponding I/O pin in the PORT group will keep its configuration. |
| 1 | The corresponding I/O pin output is driven low, or the input is connected to an internal pull-down. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.7. Data Output Value Set

**Name:** OUTSET
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to set one or more output I/O pin drive levels high, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Toggle (OUTTGL) and Data Output Value Clear (OUTCLR) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – OUTSET[31:0]** PORT Data Output Value Set
Writing '0' to a bit has no effect.
Writing '1' to a bit will set the corresponding bit in the OUT register, which sets the output drive level high for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will set the input pull direction to an internal pull-up.

| Value | Description |
|---|---|
| 0 | The corresponding I/O pin in the group will keep its configuration. |
| 1 | The corresponding I/O pin output is driven high, or the input is connected to an internal pull-up. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.8. Data Output Value Toggle

**Name:** OUTTGL
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to toggle the drive level of one or more output I/O pins, without doing a read-modify-write operation. Changes in this register will also be reflected in the Data Output Value (OUT), Data Output Value Set (OUTSET) and Data Output Value Clear (OUTCLR) registers.

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – OUTTGL[31:0]** PORT Data Output Value Toggle
Writing '0' to a bit has no effect.
Writing '1' to a bit will toggle the corresponding bit in the OUT register, which inverts the output drive level for I/O pins configured as outputs via the Data Direction register (DIR). For pins configured as inputs via Data Direction register (DIR) with pull enabled via the Pull Enable register (PULLEN), these bits will toggle the input pull direction.

| Value | Description |
|---|---|
| 0 | The corresponding I/O pin in the PORT group will keep its configuration. |
| 1 | The corresponding OUT bit value is toggled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.9. Data Input Value

**Name:** IN
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** -

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | IN[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | IN[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | IN[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | IN[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – IN[31:0]** PORT Data Input Value
These bits are cleared when the corresponding I/O pin input sampler detects a logical low level on the input pin.
These bits are set when the corresponding I/O pin input sampler detects a logical high level on the input pin.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 32.8.10. Control

**Name:** CTRL
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Tip:** The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[31:24] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[23:16] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – SAMPLING[31:0]** Input Sampling Mode
Configures the input sampling functionality of the I/O pin input samplers, for pins configured as inputs via the Data Direction register (DIR).
The input samplers are enabled and disabled in sub-groups of eight. Thus if any pins within a byte request continuous sampling, all pins in that eight pin sub-group will be continuously sampled.

| Value | Description |
|---|---|
| 0 | On demand sampling of I/O pin is enabled. |
| 1 | Continuous sampling of I/O pin is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.11. Write Configuration

**Name:** WRCONFIG
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Only

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

This Write-only register is used to configure several pins simultaneously with the same configuration and/or peripheral multiplexing.

In order to avoid side effect of non-atomic access, 8-bit or 16-bit writes to this register will have no effect. Reading this register always returns zero.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | HWSEL | WRPINCFG | | WRPMUX | PMUX[3:0] | | | |
| Access | W | W | | W | W | W | W | W |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | DRVSTR | | | | PULLEN | INEN | PMUXEN |
| Access | | W | | | | W | W | W |
| Reset | | 0 | | | | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PINMASK[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PINMASK[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – HWSEL** Half-Word Select
This bit selects the half-word field of a 32-PORT group to be reconfigured in the atomic write operation.
This bit will always read as zero.

| Value | Description |
|---|---|
| 0 | The lower 16 pins of the PORT group will be configured. |
| 1 | The upper 16 pins of the PORT group will be configured. |

**Bit 30 – WRPINCFG** Write PINCFG
This bit determines whether the atomic write operation will update the Pin Configuration register (PINCFGy) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.
Writing '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVSTR, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN, and WRCONFIG.PINMASK values.
This bit will always read as zero.

| Value | Description |
|---|---|
| 0 | The PINCFGy registers of the selected pins will not be updated. |
| 1 | The PINCFGy registers of the selected pins will be updated. |

**Bit 28 – WRPMUX** Write PMUX
This bit determines whether the atomic write operation will update the Peripheral Multiplexing register (PMUXn) or not for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits.
Writing '0' to this bit has no effect.
Writing '1' to this bit updates the pin multiplexer configuration of the selected pins with the written WRCONFIG. PMUX value.
This bit will always read as zero.

| Value | Description |
|---|---|
| 0 | The PMUXn registers of the selected pins will not be updated. |
| 1 | The PMUXn registers of the selected pins will be updated. |

**Bits 27:24 – PMUX[3:0]** Peripheral Multiplexing
These bits determine the new value written to the Peripheral Multiplexing register (PMUXn) for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPMUX bit is set.
These bits will always read as zero.

**Bit 22 – DRVSTR** Output Driver Strength Selection
This bit determines the new value written to PINCFGy.DRVSTR for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

**Bit 18 – PULLEN** Pull Enable
This bit determines the new value written to PINCFGy.PULLEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

**Bit 17 – INEN** Input Buffer Enable
This bit determines the new value written to PINCFGy.INEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

**Bit 16 – PMUXEN** Peripheral Multiplexer Enable
This bit determines the new value written to PINCFGy.PMUXEN for all pins selected by the WRCONFIG.PINMASK and WRCONFIG.HWSEL bits, when the WRCONFIG.WRPINCFG bit is set.
This bit will always read as zero.

**Bits 15:0 – PINMASK[15:0]** Pin Mask for Multiple Pin Configuration
These bits select the pins to be configured within the half-word group selected by the WRCONFIG.HWSEL bit.
These bits will always read as zero.

| Value | Description |
|---|---|
| 0 | The configuration of the corresponding I/O pin in the half-word group will be left unchanged. |
| 1 | The configuration of the corresponding I/O pin in the half-word PORT group will be updated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 32.8.12. Event Input Control

**Name:** EVCTRL
**Offset:** 0x2C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Tip:** The I/O pins are assembled in pin groups ("PORT groups") with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PORTEI3 | EVACT3[1:0] | | PID3[4:0] | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | PORTEI2 | EVACT2[1:0] | | PID2[4:0] | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | PORTEI1 | EVACT1[1:0] | | PID1[4:0] | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | PORTEI0 | EVACT0[1:0] | | PID0[4:0] | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7, 15, 23, 31 – PORTEIx** PORT Event Input Enable x [x = 3..0]

| Value | Description |
|---|---|
| 0 | The event action x (EVACTx) will not be triggered on any incoming event. |
| 1 | The event action x (EVACTx) will be triggered on any incoming event. |

**Bits 5:6, 13:14, 21:22, 29:30 – EVACTx** PORT Event Action x [x = 3..0]
These bits define the event action the PORT will perform on event input x. See also Table 32-4.

**Bits 0:4, 8:12, 16:20, 24:28 – PIDx** PORT Event Pin Identifier x [x = 3..0]
These bits define the I/O pin on which the event action will be performed, according to Table 32-5.

**Table 32-4.** PORT Event x Action ( x = [3..0] )

| Value | Name | Description |
|---|---|---|
| 0x0 | OUT | Output register of pin will be set to level of event. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 32-4.** PORT Event x Action ( x = [3..0] ) (continued)

| Value | Name | Description |
|---|---|---|
| 0x1 | SET | Set output register of pin on event. |
| 0x2 | CLR | Clear output register of pin on event. |
| 0x3 | TGL | Toggle output register of pin on event. |

**Table 32-5.** PORT Event x Pin Identifier ( x = [3..0] )

| Value | Name | Description |
|---|---|---|
| 0x0 | PIN0 | Event action to be executed on PIN 0. |
| 0x1 | PIN1 | Event action to be executed on PIN 1. |
| ... | ... | ... |
| 0x31 | PIN31 | Event action to be executed on PIN 31. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 32.8.13. Peripheral Multiplexing n

**Name:** PMUX
**Offset:** 0x30 + n*0x01 [n=0..15]
**Reset:** 0x00
**Property:** PAC Write-Protection

**Tip:** The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 16 Peripheral Multiplexing registers in each group, one for every set of two subsequent I/O lines. The n denotes the number of the set of I/O lines.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PMUXO[3:0] | | | | PMUXE[3:0] | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:4 – PMUXO[3:0]** Peripheral Multiplexing for Odd-Numbered Pin
These bits select the peripheral function for odd-numbered pins (2*n + 1) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations*.

| PMUXO[3:0] | Name | Description |
|---|---|---|
| 0x0 | A | Peripheral function A selected |
| 0x1 | B | Peripheral function B selected |
| 0x2 | C | Peripheral function C selected |
| 0x3 | D | Peripheral function D selected |
| 0x4 | E | Peripheral function E selected |
| 0x5 | F | Peripheral function F selected |
| 0x6 | G | Peripheral function G selected |
| 0x7 | H | Peripheral function H selected |
| 0x8 | I | Peripheral function I selected |
| 0x9 | J | Peripheral function J selected |
| 0xA | K | Peripheral function K selected |
| 0xB | L | Peripheral function L selected |
| 0xC | M | Peripheral function M selected |
| 0xD | N | Peripheral function N selected |
| 0xE-0xF | - | Reserved |

**Bits 3:0 – PMUXE[3:0]** Peripheral Multiplexing for Even-Numbered Pin
These bits select the peripheral function for even-numbered pins (2*n) of a PORT group, if the corresponding PINCFGy.PMUXEN bit is '1'.
Not all possible values for this selection may be valid. For more details, refer to the *I/O Multiplexing and Considerations.*

| PMUXE[3:0] | Name | Description |
|---|---|---|
| 0x0 | A | Peripheral function A selected |
| 0x1 | B | Peripheral function B selected |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**PMUXE** (continued)

| PMUXE[3:0] | Name | Description |
|---|---|---|
| 0x2 | C | Peripheral function C selected |
| 0x3 | D | Peripheral function D selected |
| 0x4 | E | Peripheral function E selected |
| 0x5 | F | Peripheral function F selected |
| 0x6 | G | Peripheral function G selected |
| 0x7 | H | Peripheral function H selected |
| 0x8 | I | Peripheral function I selected |
| 0x9 | J | Peripheral function J selected |
| 0xA | K | Peripheral function K selected |
| 0xB | L | Peripheral function L selected |
| 0xC | M | Peripheral function M selected |
| 0xD | N | Peripheral function N selected |
| 0xE-0xF | - | Reserved |

### Related Links

I/O Multiplexing and Considerations


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 32.8.14. Pin Configuration

**Name:** PINCFG
**Offset:** 0x40 + n*0x01 [n=0..31]
**Reset:** 0x40
**Property:** PAC Write-Protection

**Tip:** The I/O pins are assembled in pin groups (”PORT groups”) with up to 32 pins. Group 0 consists of the PA pins, group 1 is for the PB pins, etc. Each pin group has its own PORT registers, with a 0x80 address spacing. For example, the register address offset for the Data Direction (DIR) register for group 0 (PA00 to PA31) is 0x00, and the register address offset for the DIR register for group 1 (PB00 to PB31) is 0x80.

There are up to 32 Pin Configuration registers in each PORT group, one for each I/O line.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | DRVSTR | | | | PULLEN | INEN | PMUXEN |
| Access | | RW | | | | RW | RW | RW |
| Reset | | 1 | | | | 0 | 0 | 0 |

**Bit 6 – DRVSTR** Output Driver Strength Selection
This bit controls the output driver strength of an I/O pin configured as an output.
**Note:** The reset value for DRVSTR is 0 for PA24 and PA25 for devices with USB.

| Value | Description |
|---|---|
| 0 | Pin drive strength is set to normal drive strength. |
| 1 | Pin drive strength is set to stronger drive strength. |

**Bit 2 – PULLEN** Pull Enable
This bit enables the internal pull-up or pull-down resistor of an I/O pin configured as an input.

| Value | Description |
|---|---|
| 0 | Internal pull resistor is disabled, and the input is in a high-impedance configuration. |
| 1 | Internal pull resistor is enabled, and the input is driven to a defined logic level in the absence of external input. |

**Bit 1 – INEN** Input Enable
This bit controls the input buffer of an I/O pin configured as either an input or output.
Writing a zero to this bit disables the input buffer completely, preventing read-back of the Physical Pin state when the pin is configured as either an input or output.

| Value | Description |
|---|---|
| 0 | Input buffer for the I/O pin is disabled, and the input value will not be sampled. |
| 1 | Input buffer for the I/O pin is enabled, and the input value will be sampled when required. |

**Bit 0 – PMUXEN** Peripheral Multiplexer Enable
This bit enables or disables the peripheral multiplexer selection set in the Peripheral Multiplexing register (PMUXn) to enable or disable alternative peripheral control over an I/O pin direction and output drive value.
Writing a zero to this bit allows the PORT to control the pad direction through the Data Direction register (DIR) and output drive value through the Data Output Value register (OUT). The peripheral multiplexer value in PMUXn is ignored. Writing '1' to this bit enables the peripheral selection in PMUXn to control the pad. In this configuration, the Physical Pin state may still be read from the Data Input Value register (IN) if PINCFGn.INEN is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The peripheral multiplexer selection is disabled, and the PORT registers control the direction and output drive value. |
| 1 | The peripheral multiplexer selection is enabled, and the selected peripheral function controls the direction and output drive value. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 33. SERCOM – Serial Communication Interface

## 33.1. Overview

There are up to eight instances of the Serial Communication interface (SERCOM) peripheral.

A SERCOM can be configured to support a number of modes: $I^2C$, SPI, and USART. When an instance of SERCOM is configured and enabled, all of the resources of that SERCOM instance will be dedicated to the selected mode.

The SERCOM serial engine consists of a transmitter and receiver, baud-rate generator and address matching functionality. It can use the internal generic clock or an external clock. Using an external clock allows the SERCOM to be operated in all Sleep modes.

**Related Links**

SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
SERCOM SPI – SERCOM Serial Peripheral Interface
SERCOM I2C – Inter-Integrated Circuit
SERCOM I2C Configurations

## 33.2. Features

- Interface for Configuring into one of the following (selected by CTRLA.MODE[2:0]):
  - Inter-Integrated Circuit ($I^2C$) two-wire serial interface
  - System Management Bus (SMBus™) compatible
  - Serial Peripheral Interface (SPI)
  - Universal Synchronous/Asynchronous Receiver/Transmitter (USART)
- Single Transmit Buffer and Double Receive Buffer
- Baud-rate Generator
- Address Match/mask Logic
- Operational in all Sleep modes with an External Clock Source
- Can be used with DMA
- 32-bit Extension for Better System Bus Utilization

See the Related Links for full feature lists of the interface configurations.

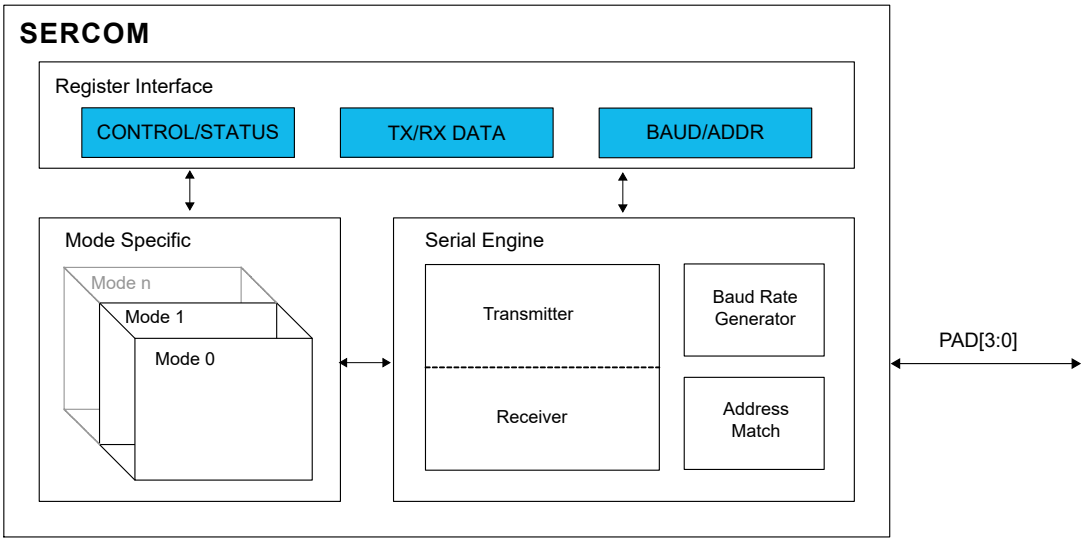
**Related Links**

SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
SERCOM SPI – SERCOM Serial Peripheral Interface
SERCOM I2C – Inter-Integrated Circuit
SERCOM I2C Configurations


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 33.3. Block Diagram

**Figure 33-1.** SERCOM Block Diagram




## 33.4. Signal Description

See the respective SERCOM mode chapters for details.

**Related Links**

SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
SERCOM SPI – SERCOM Serial Peripheral Interface
SERCOM I2C – Inter-Integrated Circuit

## 33.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 33.5.1. I/O Lines

Using the SERCOM I/O lines requires the I/O pins to be configured using port configuration (PORT).

The SERCOM has four internal pads, PAD[3:0], and the signals from $I^2C$, SPI and USART are routed through these SERCOM pads through a multiplexer. The configuration of the multiplexer is available from the different SERCOM modes. Refer to the mode specific chapters for additional information.

**Related Links**

SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
SERCOM SPI – SERCOM Serial Peripheral Interface
SERCOM I2C – Inter-Integrated Circuit
PORT - I/O Pin Controller
Block Diagram

### 33.5.2. Power Management

The SERCOM can operate in any Sleep mode provided the selected clock source is running. SERCOM interrupts can be configured to wake the device from sleep modes.

**Related Links**

PM – Power Manager


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

The SERCOM uses two generic clocks: GCLK_SERCOMx_CORE and GCLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) is required to clock the SERCOM while working as a host. The slow clock (GCLK_SERCOMx_SLOW) is only required for certain functions. See specific mode chapters for details.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SERCOM.

The generic clocks are asynchronous to the user interface clock (CLK_SERCOMx_APB). Due to this asynchronicity, writing to certain registers will require synchronization between the clock domains. Refer to Synchronization for details.

**Related Links**

GCLK - Generic Clock Controller
MCLK – Main Clock

### 33.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). The DMAC must be configured before the SERCOM DMA requests are used.

**Related Links**

DMAC – Direct Memory Access Controller

### 33.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller (NVIC). The NVIC must be configured before the SERCOM interrupts are used.

**Related Links**

Nested Vector Interrupt Controller

### 33.5.6. Events

Not applicable.

### 33.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

### 33.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
PAC - Peripheral Access Controller

### 33.5.9. Analog Connections

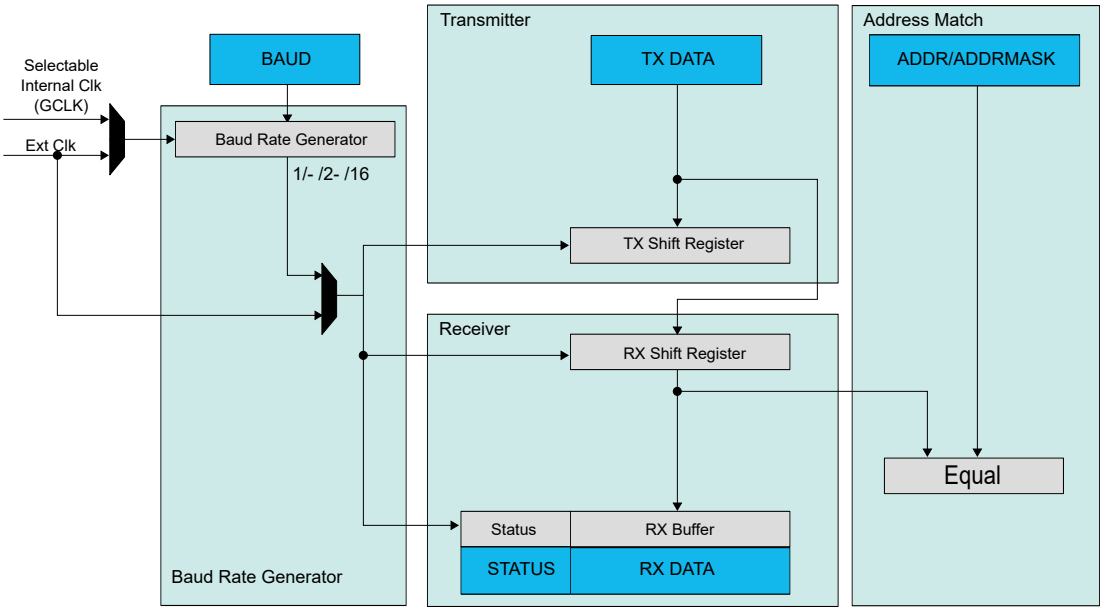
Not applicable.

## 33.6. Functional Description

### 33.6.1. Principle of Operation

The basic structure of the SERCOM serial engine is shown in *SERCOM Serial Engine*. Labels in capital letters are synchronous to the system clock and accessible by the CPU; labels in lowercase letters can be configured to run on the GCLK_SERCOMx_CORE clock or an external clock.

**Figure 33-2.** SERCOM Serial Engine




The transmitter consists of a single write buffer and a Shift register.

The receiver consists of a one-level ($I^2C$), or two-level (USART, SPI) receive buffer and a Shift register.

The baud-rate generator is capable of running on the GCLK_SERCOMx_CORE clock or an external clock.

Address matching logic is included for SPI and $I^2C$ operation.

### 33.6.2. Basic Operation

#### 33.6.2.1. Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE) as shown in the table below.

**Table 33-1.** SERCOM Modes

| CTRLA.MODE | Description |
|---|---|
| 0x0 | USART with external clock |
| 0x1 | USART with internal clock |
| 0x2 | SPI in client operation |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 33-1.** SERCOM Modes (continued)

| CTRLA.MODE | Description |
|---|---|
| 0x3 | SPI in host operation |
| 0x4 | $I^2C$ client operation |
| 0x5 | $I^2C$ host operation |
| 0x6-0x7 | Reserved |

For further initialization information, see the respective SERCOM mode chapters:

**Related Links**

SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
SERCOM SPI – SERCOM Serial Peripheral Interface
SERCOM I2C – Inter-Integrated Circuit

### 33.6.2.2. Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

### 33.6.2.3. Clock Generation – Baud-Rate Generator

The baud-rate generator, as shown in Figure 33-3, generates internal clocks for asynchronous and synchronous communication. The output frequency ($f_{BAUD}$) is determined by the Baud register (BAUD) setting and the baud reference frequency ($f_{ref}$). The baud reference clock is the serial engine clock, and it can be internal or external.

For asynchronous communication, the /16 (divide-by-16) output is used when transmitting, whereas the /1 (divide-by-1) output is used while receiving.

For synchronous communication, the /2 (divide-by-2) output is used.

This functionality is automatically configured, depending on the selected operating mode.

**Figure 33-3.** Baud Rate Generator

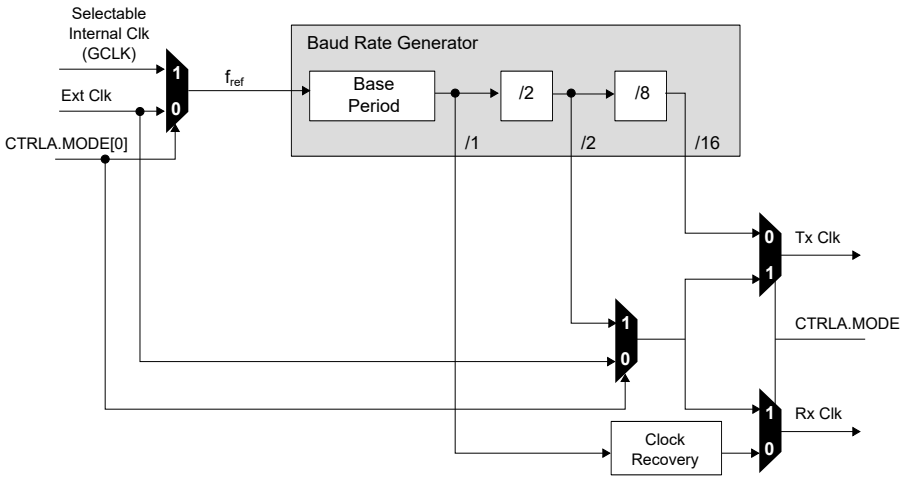



Table 33-2 contains equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For asynchronous operation, there is one mode: *arithmetic mode*, the BAUD register value is 16 bits (0 to 65,535).*fractional mode*, the BAUD register value is 13 bits, while the fractional adjustment is 3 bits. In this mode the BAUD setting must be greater than or equal to 1.

For synchronous operation, the BAUD register value is 8 bits (0 to 255).

**Table 33-2.** Baud Rate Equations

| Operating Mode | Condition | Baud Rate (Bits Per Second) | BAUD Register Value Calculation |
|---|---|---|---|
| Asynchronous Arithmetic | $f_{BAUD} \leq \frac{f_{ref}}{16}$ | $f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$ | $BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}}\right)$ |
| Asynchronous Fractional | $f_{BAUD} \leq \frac{f_{ref}}{S}$ | $f_{BAUD} = \frac{f_{ref}}{S \cdot \left(BAUD + \frac{FP}{8}\right)}$ | $BAUD = \frac{f_{ref}}{S \cdot f_{BAUD}} - \frac{FP}{8}$ |
| Synchronous | $f_{BAUD} \leq \frac{f_{ref}}{2}$ | $f_{BAUD} = \frac{f_{ref}}{2 \cdot (BAUD + 1)}$ | $BAUD = \frac{f_{ref}}{2 \cdot f_{BAUD}} - 1$ |

*S* - Number of samples per bit, which can be 16, 8, or 3.

The Asynchronous Fractional option is used for auto-baud detection.

The baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

### 33.6.2.3.1. Asynchronous Arithmetic Mode BAUD Value Selection

The formula given for $f_{BAUD}$ calculates the average frequency over 65536 $f_{ref}$ cycles. Although the BAUD register can be set to any value between 0 and 65536, the actual average frequency of $f_{BAUD}$ over a single frame is more granular. The BAUD register values that will affect the average frequency over a single frame lead to an integer increase in the cycles per frame (CPF)

$$CPF = \frac{f_{ref}}{f_{BAUD}}(D + S)$$

where

- *D* represent the data bits per frame
- *S* represent the sum of start and first stop bits, if present.

Table 33-3 shows the BAUD register value versus baud frequency $f_{BAUD}$ at a serial engine frequency of 48 MHz. This assumes a *D* value of 8 bits and an *S* value of 2 bits (10 bits, including start and stop bits).

**Table 33-3.** BAUD Register Value vs. Baud Frequency

| BAUD Register Value | Serial Engine CPF | $f_{BAUD}$ at 100MHz Serial Engine Frequency ($f_{REF}$) |
|---|---|---|
| 0 – 406 | 161 | 6.211 MHz |
| 407 – 808 | 162 | 6.211 MHz |
| 809 – 1205 | 163 | 6.173 MHz |
| ... | ... | ... |
| 65206 | 31775 | 31.47 kHz |
| 65207 | 31872 | 31.38 kHz |
| 65208 | 31969 | 31.28 kHz |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 33.6.3. Additional Features

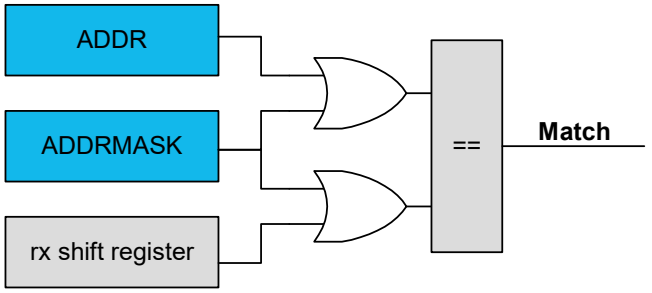
### 33.6.3.1. Address Match and Mask

The SERCOM address match and mask feature is capable of matching either one address, two unique addresses, or a range of addresses with a mask, based on the mode selected. The match uses seven or eight bits, depending on the mode.

#### 33.6.3.1.1. Address With Mask

An address written to the Address bits in the Address register (ADDR.ADDR), and a mask written to the Address Mask bits in the Address register (ADDR.ADDRMASK) will yield an address match. All bits that are masked are not included in the match. Note that writing the ADDR.ADDRMASK to 'all zeros' will match a single unique address, while writing ADDR.ADDRMASK to 'all ones' will result in all addresses being accepted.

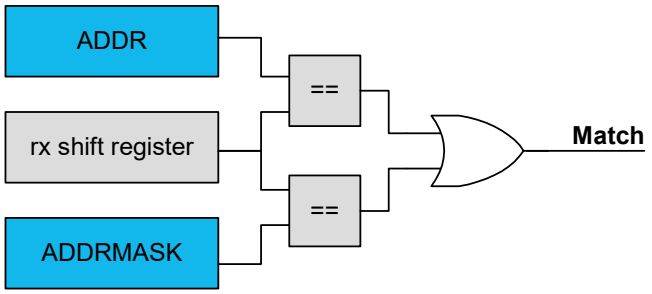
**Figure 33-4.** Address With Mask




#### 33.6.3.1.2. Two Unique Addresses

The two addresses written to ADDR and ADDRMASK will cause a match.

**Figure 33-5.** Two Unique Addresses




#### 33.6.3.1.3. Address Range

The range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK will cause a match. ADDR.ADDR and ADDR.ADDRMASK can be set to any two addresses, with ADDR.ADDR acting as the upper limit and ADDR.ADDRMASK acting as the lower limit.

**Figure 33-6.** Address Range




## 33.6.4. DMA Operation

The available DMA interrupts and their depend on the operation mode of the SERCOM peripheral. Refer to the Functional Description sections of the respective SERCOM mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter
SERCOM SPI – SERCOM Serial Peripheral Interface
SERCOM I2C – Inter-Integrated Circuit

### 33.6.5. Interrupts

Interrupt sources are mode specific. See the respective SERCOM mode chapters for details.

Each interrupt source has its own Interrupt flag.

The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met.

Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the Interrupt flag is cleared, the interrupt is disabled, or the SERCOM is reset. For details on clearing Interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which Interrupt condition occurred. The user must read the INTFLAG register to determine which Interrupt condition is present.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

**Related Links**
Nested Vector Interrupt Controller

### 33.6.6. Events

Not applicable.

### 33.6.7. Sleep Mode Operation

The peripheral can operate in any Sleep mode where the selected serial clock is running. This clock can be external or generated by the internal baud-rate generator.

The SERCOM interrupts can be used to wake-up the device from Sleep modes. Refer to the different SERCOM mode chapters for details.

### 33.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

**Related Links**
Register Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 34. SERCOM USART - SERCOM Synchronous and Asynchronous Receiver and Transmitter

## 34.1. Overview

The Universal Synchronous and Asynchronous Receiver and Transmitter (USART) is one of the available modes in the Serial Communication Interface (SERCOM).

The USART uses the SERCOM transmitter and receiver, see Block Diagram. Labels in uppercase letters are synchronous to CLK_SERCOMx_APB and accessible for CPU. Labels in lowercase letters can be programmed to run on the internal generic clock or an external clock.

The transmitter consists of a single write buffer, a Shift register, and control logic for different frame formats. The write buffer support data transmission without any delay between frames. The receiver consists of a two-level receive buffer and a Shift register. Status information of the received data is available for error checking. Data and clock recovery units ensure robust synchronization and noise filtering during asynchronous data reception.

**Related Links**

SERCOM – Serial Communication Interface

## 34.2. USART Features

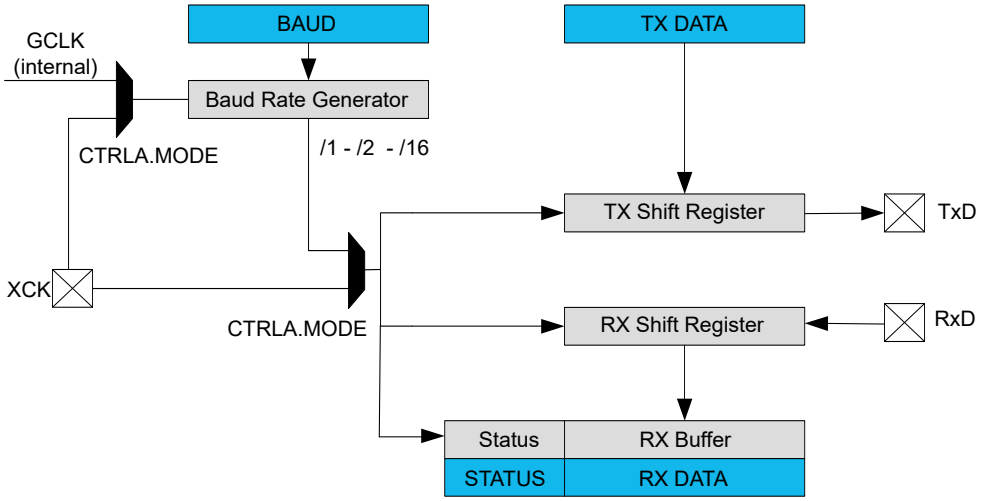
- Full-duplex Operation
- Asynchronous (with Clock Reconstruction) or Synchronous Operation
- Internal or External Clock source for Asynchronous and Synchronous Operation
- Baud-rate Generator
- Supports Serial Frames with 5, 6, 7, 8 or 9 Data bits and 1 or 2 Stop bits
- Odd or Even Parity Generation and Parity Check
- Selectable LSB- or MSB-first Data Transfer
- Buffer Overflow and Frame Error Detection
- Noise Filtering, Including False Start bit Detection and Digital Low-pass Filter
- Collision Detection
- Can Operate in all Sleep modes
- Operation at Speeds up to Half the System Clock for Internally Generated Clocks
- Operation at Speeds up to the System Clock for Externally Generated Clocks
- RTS and CTS Flow Control
- IrDA Modulation and Demodulation up to 115.2kbps
- LIN Host Support
- LIN Client Support
  - Auto-baud and break character detection
- ISO 7816 T=0 or T=1 protocols for Smart Card Interfacing
- RS485 Support
- Start-of-frame detection
- Two-Level Receive Buffer
- Can work with DMA
- 32-bit Extension for Better System Bus Utilization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Features

## 34.3. Block Diagram

**Figure 34-1.** USART Block Diagram




## 34.4. Signal Description

**Table 34-1.** SERCOM USART Signals

| Signal Name | Type | Description |
|---|---|---|
| PAD[3:0] | Digital I/O | General SERCOM pins |

One signal can be mapped to one of several pins.

**Related Links**
I/O Multiplexing and Considerations

## 34.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 34.5.1. I/O Lines

Using the USART's I/O lines requires the I/O pins to be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in USART mode, the SERCOM controls the direction and value of the I/O pins according to the table below. If the receiver or transmitter is disabled, these pins can be used for other purposes.

**Table 34-2.** USART Pin Configuration

| Pin | Pin Configuration |
|---|---|
| TxD | Output |
| RxD | Input |
| XCK | Output or input |

The combined configuration of PORT and the Transmit Data Pinout and Receive Data Pinout bit fields in the Control A register (CTRLA.TXPO and CTRLA.RXPO, respectively) will define the physical position of the USART signals in Table 34-2.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
PORT - I/O Pin Controller

### 34.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

**Related Links**
PM – Power Manager

### 34.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SERCOMx_CORE. This clock must be configured and enabled in the Generic Clock Controller before using the SERCOMx_CORE. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writing to certain registers will require synchronization to the clock domains. Refer to *Synchronization* for further details.

**Related Links**
Peripheral Clock Masking
Synchronization
GCLK - Generic Clock Controller

### 34.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

**Related Links**
DMAC – Direct Memory Access Controller

### 34.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller

### 34.5.6. Events

Not applicable.

### 34.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

**Related Links**
DBGCTRL
Debug Control


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.5.8. Register Access Protection

Registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC).

PAC write protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 34.5.9. Analog Connections

Not applicable.

## 34.6. Functional Description

### 34.6.1. Principle of Operation

The USART uses the following lines for data transfer:

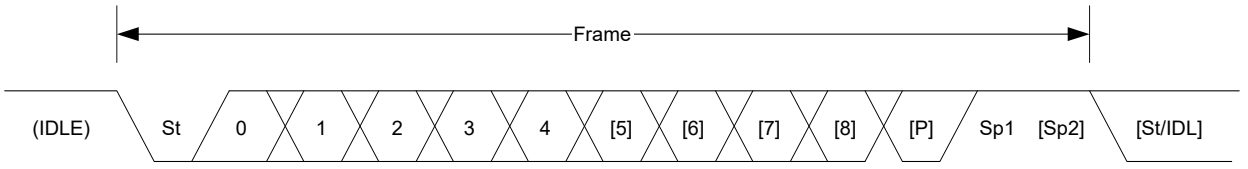
- RxD for receiving
- TxD for transmitting
- XCK for the transmission clock in synchronous operation

USART data transfer is frame based. A serial frame consists of:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- No, even or odd parity bit
- 1 or 2 stop bits

A frame starts with the Start bit followed by one character of Data bits. If enabled, the parity bit is inserted after the Data bits and before the first Stop bit. After the stop bit(s) of a frame, either the next frame can follow immediately, or the communication line can return to the Idle (high) state. The figure below illustrates the possible frame formats. Brackets denote optional bits.

**Figure 34-2.** Frame Formats




**St** Start bit. Signal is always low.

**n, [n]** Data bits. 0 to [5..9]

**[P]** Parity bit. Either odd or even.

**Sp, [Sp]** Stop bit. Signal is always high.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**IDLE** No frame is transferred on the communication line. Signal is always high in this state.

## 34.6.2. Basic Operation

### 34.6.2.1. Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (CTRL.ENABLE=0):

- Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits.
- Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits.
- Baud register (BAUD)

When the USART is enabled or is being enabled (CTRLA.ENABLE=1), any writing attempt to these registers will be discarded. If the peripheral is being disabled, writing to these registers will be executed after disabling is completed. Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the USART is enabled, it must be configured by these steps:

1. Select either external (0x0) or internal clock (0x1) by writing the Operating Mode value in the CTRLA register (CTRLA.MODE).
2. Select either Asynchronous (0) or Synchronous (1) Communication mode by writing the Communication Mode bit in the CTRLA register (CTRLA.CMODE).
3. Select pin for receive data by writing the Receive Data Pinout value in the CTRLA register (CTRLA.RXPO).
4. Select pads for the transmitter and external clock by writing the Transmit Data Pinout bit in the CTRLA register (CTRLA.TXPO).
5. Configure the Character Size field in the CTRLB register (CTRLB.CHSIZE) for character size.
6. Set the Data Order bit in the CTRLA register (CTRLA.DORD) to determine MSB- or LSB-first data transmission.
7. To use parity mode:
   a. Enable Parity mode by writing 0x1 to the Frame Format field in the CTRLA register (CTRLA.FORM).
   b. Configure the Parity Mode bit in the CTRLB register (CTRLB.PMODE) for even or odd parity.
8. Configure the number of stop bits in the Stop Bit Mode bit in the CTRLB register (CTRLB.SBMODE).
9. When using an internal clock, write the Baud register (BAUD) to generate the desired baud rate.
10. Enable the transmitter and receiver by writing '1' to the Receiver Enable and Transmitter Enable bits in the CTRLB register (CTRLB.RXEN and CTRLB.TXEN).

### 34.6.2.2. Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

### 34.6.2.3. Clock Generation and Selection

For both Synchronous and Asynchronous modes, the clock used for shifting and sampling data can be generated internally by the SERCOM baud-rate generator or supplied externally through the XCK line.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The Synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the Asynchronous mode is selected by writing a zero to CTRLA.CMODE.

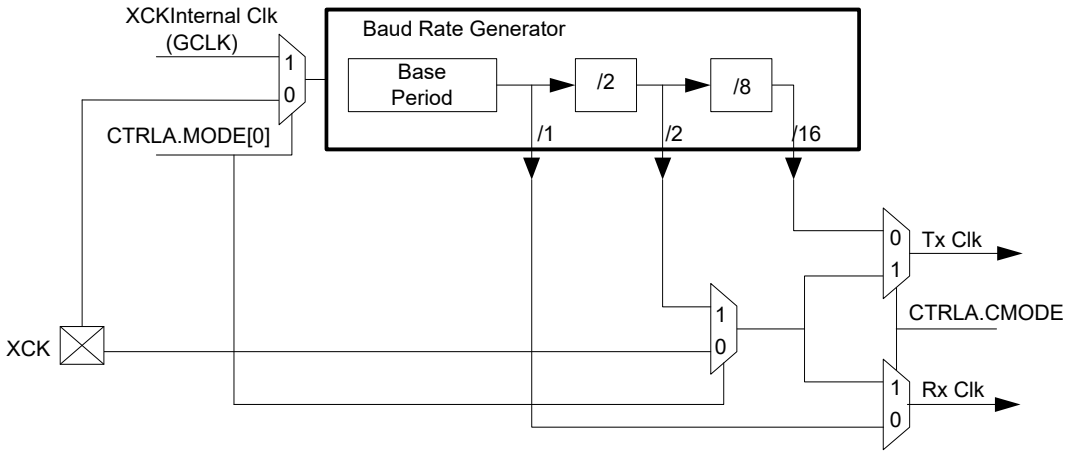
The internal clock source is selected by writing 0x1 to the Operation Mode bit field in the Control A register (CTRLA.MODE), the external clock source is selected by writing 0x0 to CTRLA.MODE.

The SERCOM baud-rate generator is configured as in the figure below.

In Asynchronous mode (CTRLA.CMODE=0), the 16-bit Baud register value is used.

In Synchronous mode (CTRLA.CMODE=1), the eight LSBs of the Baud register are used. Refer to *Clock Generation – Baud-Rate Generator* for details on configuring the baud rate.

**Figure 34-3.** Clock Generation




**Related Links**

Clock Generation – Baud-Rate Generator
Asynchronous Arithmetic Mode BAUD Value Selection

#### 34.6.2.3.1. Synchronous Clock Operation

In Synchronous mode, the CTRLA.MODE bit field determines whether the transmission clock line (XCK) serves either as input or output. The dependency between clock edges, data sampling, and data change is the same for internal and external clocks. Data input on the RxD pin is sampled at the opposite XCK clock edge when data is driven on the TxD pin.

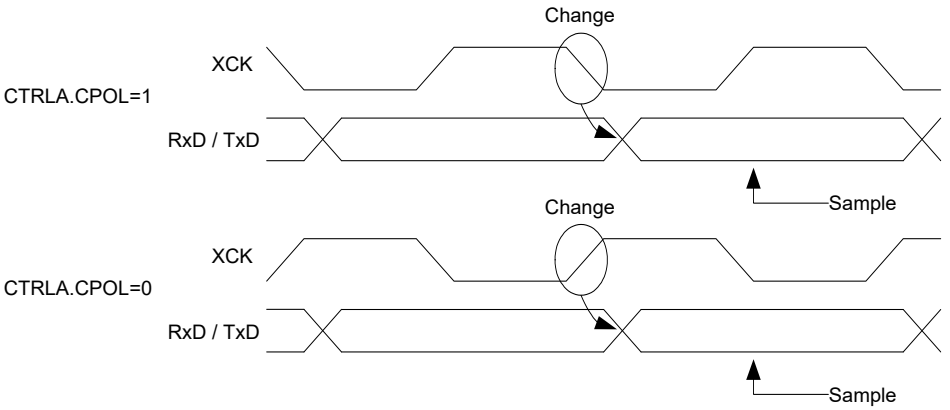
The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling, and which is used for TxD change:

When CTRLA.CPOL is '0', the data will be changed on the rising edge of XCK, and sampled on the falling edge of XCK.

When CTRLA.CPOL is '1', the data will be changed on the falling edge of XCK, and sampled on the rising edge of XCK.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 34-4.** Synchronous Mode XCK Timing




When the clock is provided through XCK (CTRLA.MODE=0x0), the Shift registers operate directly on the XCK clock. This means that XCK is not synchronized with the system clock and, therefore, can operate at frequencies up to the system frequency.

#### 34.6.2.4. Data Register

The USART Transmit Data register (TxDATA) and USART Receive Data register (RxDATA) share the same I/O address, referred to as the Data register (DATA). Writing the DATA register will update the TxDATA register. Reading the DATA register will return the contents of the RxDATA register.

#### 34.6.2.5. Data Transmission

Data transmission is initiated by writing the data to be sent into the DATA register. Then, the data in TxDATA will be moved to the Shift register when the Shift register is empty and ready to send a new frame. After the Shift register is loaded with data, the data frame will be transmitted.

When the entire data frame including Stop bit has been transmitted (both the Tx buffer and the shift register are empty) and no new data was written to the DATA register, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set, and the optional interrupt can be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

##### 34.6.2.5.1. Disabling the Transmitter

The transmitter is disabled by writing '0' to the Transmitter Enable bit in the CTRLB register (CTRLB.TXEN).

Disabling the transmitter will complete only after any ongoing and pending transmissions are completed, i.e., there is no data in the Transmit Shift register and TxDATA to transmit.

#### 34.6.2.6. Data Reception

The receiver accepts data when a valid Start bit is detected. Each bit following the Start bit will be sampled according to the baud rate or XCK clock, and shifted into the receive Shift register until the first Stop bit of a frame is received. The second Stop bit will be ignored by the receiver.

When the first Stop bit is received and a complete serial frame is present in the Receive Shift register, the contents of the Shift register will be moved into the two-level receive buffer. Then, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set, and the optional interrupt can be generated.

The received data can be read from the DATA register when the Receive Complete Interrupt flag is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 34.6.2.6.1. Disabling the Receiver

Writing '0' to the Receiver Enable bit in the CTRLB register (CTRLB.RXEN) will disable the receiver, flush the two-level receive buffer, and data from ongoing receptions will be lost.

#### 34.6.2.6.2. Error Bits

The USART receiver has three error bits in the Status (STATUS) register: Frame Error (FERR), Buffer Overflow (BUFOVF), and Parity Error (PERR). Once an error happens, the corresponding error bit will be set until it is cleared by writing '1' to it. These bits are also cleared automatically when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the Immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

When CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA, until the Receiver Complete Interrupt flag (INTFLAG.RXC) is cleared.

When CTRLA.IBON=0, the Buffer Overflow condition is attending data through the receive FIFO, which will then set the INTFLAG.ERROR bit. After the received data is read, STATUS.BUFOVF (and INTFLAG.ERROR ) will be set along with INTFLAG.RXC.

#### 34.6.2.6.3. Asynchronous Data Reception

The USART includes a clock recovery and data recovery unit for handling asynchronous data reception.

The clock recovery logic can synchronize the incoming asynchronous serial frames at the RxD pin to the internally generated baud-rate clock.

The data recovery logic samples and applies a low-pass filter to each incoming bit, thereby improving the noise immunity of the receiver.

#### 34.6.2.6.4. Asynchronous Operational Range

The operational range of the asynchronous reception depends on the accuracy of the internal baud-rate clock, the rate of the incoming frames, and the frame size (in number of bits). In addition, the operational range of the receiver is depending on the difference between the received bit rate and the internally generated baud rate. If the baud rate of an external transmitter is too high or too low compared to the internally generated baud rate, the receiver will not be able to synchronize the frames to the start bit.

There are two possible sources for a mismatch in baud rate: First, the reference clock will always have some minor instability. Second, the baud-rate generator cannot always do an exact division of the reference clock frequency to get the baud rate desired. In this case, the BAUD register value should be set to give the lowest possible error. Refer to *Clock Generation – Baud-Rate Generator* for details.

Recommended maximum receiver baud-rate errors for various character sizes are shown in the table below.

**Table 34-3.** Asynchronous Receiver Error for 16-fold Oversampling

| D (Data bits+Parity) | $R_{SLOW}$ [%] | $R_{FAST}$ [%] | Max. total error [%] | Recommended max. Rx error [%] |
|---|---|---|---|---|
| 5 | 94.12 | 107.69 | +5.88/-7.69 | ±2.5 |
| 6 | 94.92 | 106.67 | +5.08/-6.67 | ±2.0 |
| 7 | 95.52 | 105.88 | +4.48/-5.88 | ±2.0 |
| 8 | 96.00 | 105.26 | +4.00/-5.26 | ±2.0 |
| 9 | 96.39 | 104.76 | +3.61/-4.76 | ±1.5 |
| 10 | 96.70 | 104.35 | +3.30/-4.35 | ±1.5 |

The following equations calculate the ratio of the incoming data rate and internal receiver baud rate:

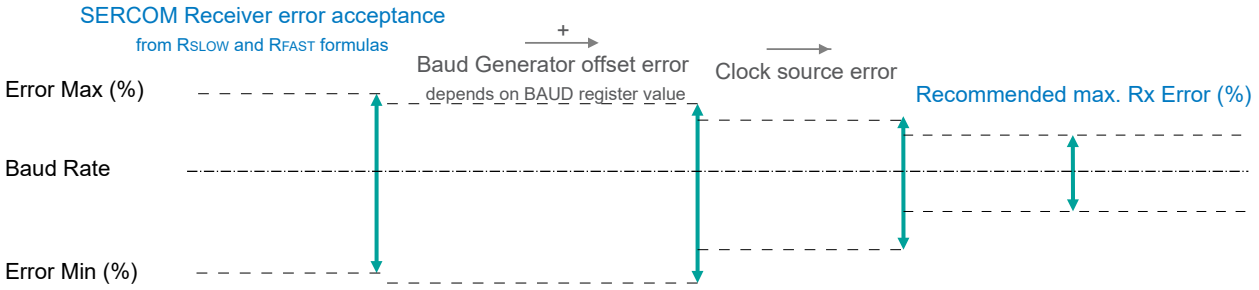

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

$$R_{SLOW} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \quad R_{FAST} = \frac{(D+2)S}{(D+1)S+S_M}$$

- $R_{SLOW}$ is the ratio of the slowest incoming data rate that can be accepted in relation to the receiver baud rate
- $R_{FAST}$ is the ratio of the fastest incoming data rate that can be accepted in relation to the receiver baud rate
- $D$ is the sum of character size and parity size ($D$ = 5 to 10 bits)
- $S$ is the number of samples per bit ($S$ = 16, 8 or 3)
- $S_F$ is the first sample number used for majority voting ($S_F$ = 7, 3, or 2) when CTRLA.SAMPA=0.
- $S_M$ is the middle sample number used for majority voting ($S_M$ = 8, 4, or 2) when CTRLA.SAMPA=0.

The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error. Its connection to the SERCOM Receiver error acceptance is depicted in this figure:

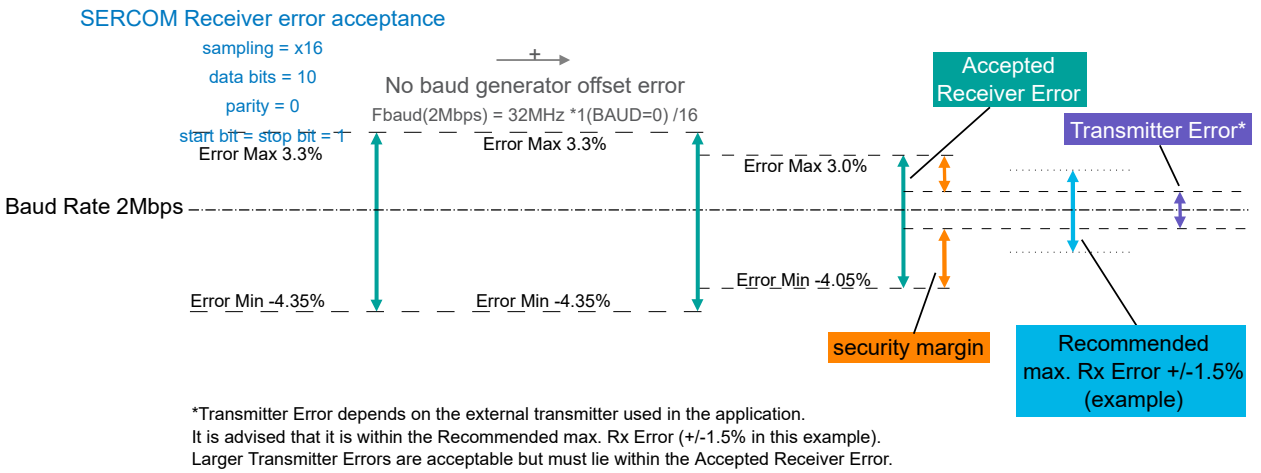
**Figure 34-5.** USART Rx Error Calculation




The recommendation values in the table above accommodate errors of the clock source and the baud generator. The following figure gives an example for a baud rate of 3Mbps:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 34-6.** USART Rx Error Calculation Example




**Related Links**

Clock Generation – Baud-Rate Generator

Asynchronous Arithmetic Mode BAUD Value Selection

## 34.6.3. Additional Features

### 34.6.3.1. Parity

Even or odd parity can be selected for error checking by writing 0x1 to the Frame Format bit field in the Control A register (CTRLA.FORM).

If *even parity* is selected (CTRLB.PMODE=0), the Parity bit of an outgoing frame is '1' if the data contains an odd number of bits that are '1', making the total number of '1' even.

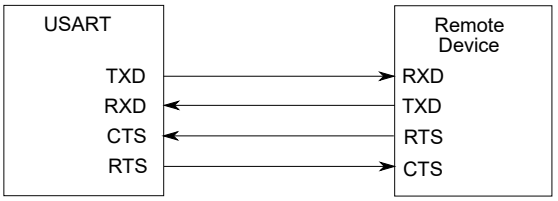
If *odd parity* is selected (CTRLB.PMODE=1), the Parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

When parity checking is enabled, the parity checker calculates the parity of the data bits in incoming frames and compares the result with the Parity bit of the corresponding frame. If a parity error is detected, the Parity Error bit in the Status register (STATUS.PERR) is set.

### 34.6.3.2. Hardware Handshaking

The USART features an out-of-band hardware handshaking flow control mechanism, implemented by connecting the RTS and CTS pins with the remote device, as shown in the figure below.

**Figure 34-7.** Connection with a Remote Device for Hardware Handshaking




Hardware handshaking is only available in the following configuration:

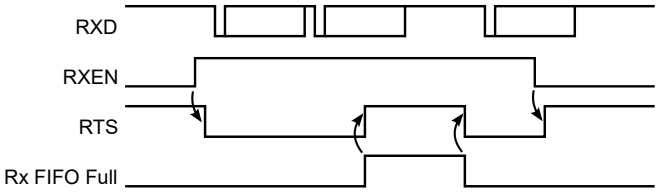
- USART with internal clock (CTRLA.MODE=1),


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Asynchronous mode (CTRLA.CMODE=0), and
- Flow control pinout (CTRLA.TXPO=2).

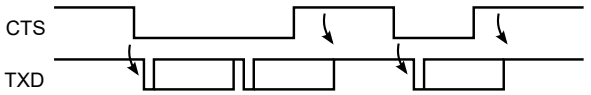
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. When the receive FIFO goes full, RTS will be set immediately and the frame being received will be stored in the Shift register until the receive FIFO is no longer full.

**Figure 34-8.** Receiver Behavior when Operating with Hardware Handshaking




The current CTS Status is in the STATUS register (STATUS.CTS). Character transmission will start only if STATUS.CTS=0. When CTS is set, the transmitter will complete the ongoing transmission and stop transmitting.

**Figure 34-9.** Transmitter Behavior when Operating with Hardware Handshaking



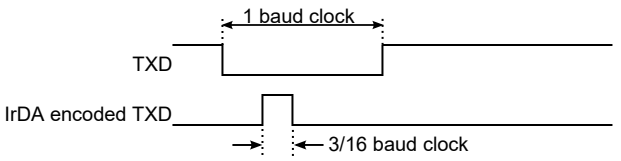

### 34.6.3.3. IrDA Modulation and Demodulation

Transmission and reception can be encoded IrDA compliant up to 115.2 kb/s. IrDA modulation and demodulation work in the following configuration:

- IrDA encoding enabled (CTRLB.ENC=1),
- Asynchronous mode (CTRLA.CMODE=0),
- and 16x sample rate (CTRLA.SAMPR[0]=0).

During transmission, each low bit is transmitted as a high pulse. The pulse width is 3/16 of the baud rate period, as illustrated in the figure below.

**Figure 34-10.** IrDA Transmit Encoding




The reception decoder has two main functions.

The first is to synchronize the incoming data to the IrDA baud rate counter. Synchronization is performed at the start of each zero pulse.

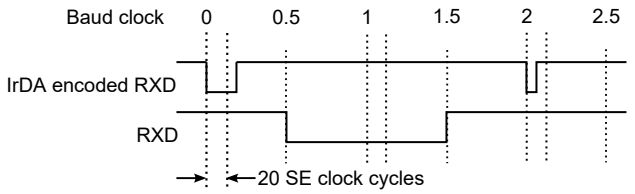
The second main function is to decode incoming Rx data. If a pulse width meets the minimum length set by configuration (RXPL.RXPL), it is accepted. When the baud rate counter reaches its middle value (1/2 bit length), it is transferred to the receiver.

**Note:** Note that the polarity of the transmitter and receiver are opposite: During transmission, a '0' bit is transmitted as a '1' pulse. During reception, an accepted '0' pulse is received as a '0' bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Example:** The figure below illustrates reception where RXPL.RXPL is set to 19. This indicates that the pulse width should be at least 20 SE clock cycles. When using BAUD=0xE666 or 160 SE cycles per bit, this corresponds to 2/16 baud clock as minimum pulse width required. In this case the first bit is accepted as a '0', the second bit is a '1', and the third bit is also a '1'. A low pulse is rejected since it does not meet the minimum requirement of 2/16 baud clock.

**Figure 34-11.** IrDA Receive Decoding




### 34.6.3.4. Break Character Detection and Auto-Baud/LIN Client

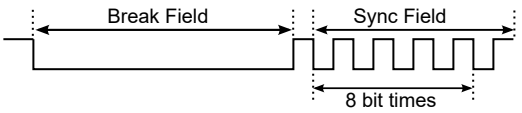
Break character detection and auto-baud are available in this configuration:

- Auto-baud frame format (CTRLA.FORM = 0x04 or 0x05),
- Asynchronous mode (CTRLA.CMODE = 0),
- and 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1).

The USART uses a break detection threshold of greater than 11 nominal bit times at the configured baud rate. At any time, if more than 11 consecutive dominant bits are detected on the bus, the USART detects a Break Field. When a break field has been detected, the Receive Break Interrupt Flag (INTFLAG.RXBRK) is set and the USART expects the sync field character to be 0x55. This field is used to update the actual baud rate in order to stay synchronized. If the received sync character is not 0x55, then the Inconsistent Sync Field error flag (STATUS.ISF) is set along with the Error Interrupt Flag (INTFLAG.ERROR), and the baud rate is unchanged.

The auto-baud follows the LIN format. All LIN Frames start with a Break Field followed by a Sync Field.

**Figure 34-12.** LIN Break and Sync Fields




After a break field is detected and the Start bit of the sync field is detected, a counter is started. The counter is then incremented for the next 8 bit times of the sync field. At the end of these 8 bit times, the counter is stopped. At this moment, the 13 Most Significant bits of the counter (value divided by 8) give the new clock divider (BAUD.BAUD), and the 3 Least Significant bits of this value (the remainder) give the new Fractional Part (BAUD.FP).

When the sync field has been received, the clock divider (BAUD.BAUD) and the Fractional Part (BAUD.FP) are updated after a synchronization delay. After the break and sync fields are received, multiple characters of data can be received.

### 34.6.3.5. LIN Host

LIN Host is available with the following configuration:

- LIN Host format (CTRLA.FORM = 0x02)
- Asynchronous mode (CTRLA.CMODE = 0)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1)

LIN frames start with a header transmitted by the Host. The header consists of the break, sync, and identifier fields. After the Host transmits the header, the addressed Client will respond with 1-8 bytes of data plus checksum.

**Figure 34-13.** LIN Frame Format




Using the LIN command field (CTRLB.LINCMD), the complete header can be automatically transmitted, or software can control transmission of the various header components.

When CTRLB.LINCMD = 0x1, software controls transmission of the LIN header. In this case, software uses the following sequence.

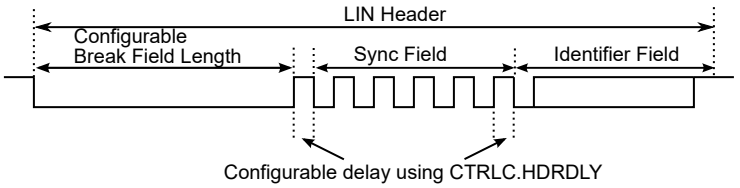
- CTRLB.LINCMD is written to 0x1.
- The DATA register written to 0x00. This triggers transmission of the break field by hardware. Writing the DATA register with any other value will also result in the transmission of the break field by hardware.
- The DATA register written to 0x55. The 0x55 value (sync) is transmitted.
- The DATA register written to the identifier. The identifier is transmitted.
  **Note:** It is recommended to use The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) to ensure no extra delay is added during the transmission of the data.

When CTRLB.LINCMD = 0x2, hardware controls transmission of the LIN header. In this case, software uses the following sequence:

- CTRLB.LINCMD is written to 0x2.
- The DATA register written to the identifier. This triggers transmission of the complete header by hardware. First the break field is transmitted. Next, the sync field is transmitted, and finally the identifier is transmitted.

In LIN Host mode, the length of the break field is programmable using the break length field (CTRLC.BRKLEN). When the LIN header command is used (CTRLB.LINCMD = 0x2), the delay between the break and sync fields, in addition to the delay between the sync and ID fields are configurable using the header delay field (CTRLC.HDRDLY). When manual transmission is used (CTRLB.LINCMD = 0x1), software controls the delay between break and sync.

**Figure 34-14.** LIN Header Generation




After header transmission is complete, the Client responds with 1-8 data bytes plus checksum.

### 34.6.3.6. RS485

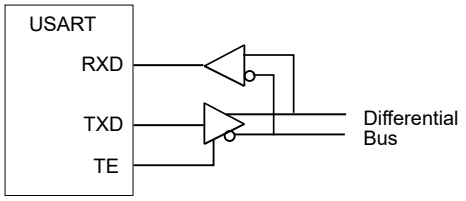
RS485 is available with the following configuration:

- USART frame format (CTRLA.FORM = 0x00 or 0x01)
- RS485 pinout (CTRLA.TXPO=0x3).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The RS485 feature enables control of an external line driver as shown in the figure below. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

**Figure 34-15.** RS485 Bus Connection




The TE pin will remain high for the complete frame including stop bit(s). If a Guard Time is programmed in the Control C register (CTRLC.GTIME), the line will remain driven after the last character completion. The following figure shows a transfer with one stop bit and CTRLC.GTIME=3.

**Figure 34-16.** Example of TE Drive with Guard Time




The Transmit Complete interrupt flag (INTFLAG.TXC) will be raised after the guard time is complete and TE goes low.

### 34.6.3.7. ISO 7816 for Smart Card Interfacing

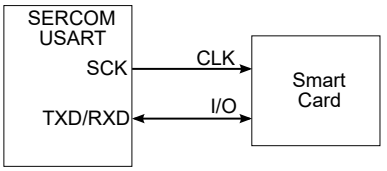
The SERCOM USART features an ISO/IEC 7816-compatible operating mode. This mode permits interfacing with smart cards and Security Access Modules (SAM) communicating through an ISO 7816 link. Both T=0 and T=1 protocols defined by the ISO 7816 specification are supported.

ISO 7816 is available with the following configuration:

- ISO 7816 format (CTRLA.FORM = 0x07)
- Inverse transmission and reception (CTRLA.RXINV=1 and CTRLA.TXINV=1)
- Single bidirectional data line (CTRLA.TXPO and CTRLA.RXPO configured to use the same data pin)
- Even parity (CTRLB.PMODE=0)
- 8-bit character size (CTRLB.CHSIZE=0)
- T=0 (CTRLA.CMODE=1) or T=1 (CTRLA.CMODE=0)

ISO 7816 is a half duplex communication on a single bidirectional line. The USART connects to a smart card as shown below. The output is only driven when the USART is transmitting. The USART is considered as the Host of the communication as it generates the clock.

**Figure 34-17.** Connection of a Smart Card to the SERCOM USART




ISO 7816 characters are specified as 8 bits with even parity. The USART must be configured accordingly.

The USART cannot operate concurrently in both receiver and transmitter modes as the communication is unidirectional. It has to be configured according to the required mode by enabling


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

or disabling either the receiver or the transmitter as desired. Enabling both the receiver and the transmitter at the same time in ISO 7816 mode may lead to unpredictable results.

The ISO 7816 specification defines an inverse transmission format. Data bits of the character must be transmitted on the I/O line at their negative value (CTRLA.RXINV=1 and CTRLA.TXINV=1).

### Protocol T=0

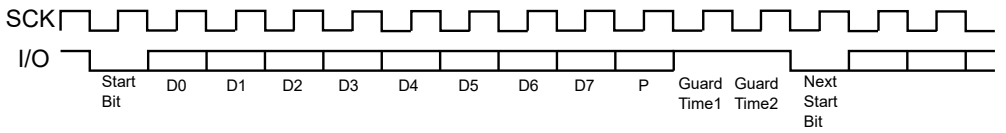
In T=0 protocol, a character is made up of:

- one start bit
- eight data bits
- one parity bit
- one guard time, which lasts two bit times

To transmit data in ISO7816, T=0 protocol mode, the USART is configured in CTRLA.CMODE =1 communication mode. The transmitter shifts out the bits and does not drive the I/O line during the guard time. Additional guard time can be added by programming the Guard Time (CTRLC.GTIME).

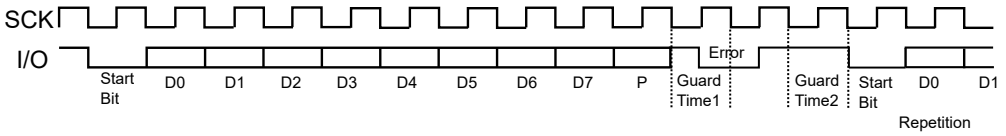
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the figure below.

**Figure 34-18.** T=0 Protocol without Parity Error




If a parity error is detected by the receiver, it drives the I/O line to 0 during the guard time, as shown in the next figure. This error bit is also named NACK, for Non Acknowledge. In this case, the character lasts 1 bit time more, as the guard time length is the same and is added to the error bit time, which lasts 1 bit time.

**Figure 34-19.** T=0 Protocol with Parity Error




When the USART is the receiver and it detects a parity error, the parity error bit in the Status Register (STATUS.PERR) is set and the character is not written to the receive FIFO.

### Receive Error Counter

The receiver also records the total number of errors (receiver parity errors and NACKs from the remote transmitter) up to a maximum of 255. This can be read in the Receive Error Count (RXERRCNT) register. RXERRCNT is automatically cleared on read.

### Receive NACK Inhibit

The receiver can also be configured to inhibit error generation. This can be achieved by setting the Inhibit Not Acknowledge (CTRLC.INACK) bit. If CTRLC.INACK is 1, no error signal is driven on the I/O line even if a parity error is detected. Moreover, if CTRLC.INACK is set, the erroneous received character is stored in the receive FIFO, and the STATUS.PERR bit is set. Inhibit not acknowledge (CTRLC.INACK) takes priority over disable successive receive NACK (CTRLC.DSNACK).

### Transmit Character Repetition

When the USART is transmitting a character and gets a NACK, it can automatically repeat the character before moving on to the next character. Repetition is enabled by writing the Maximum


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Iterations register (CTRLC.MAXITER) to a non-zero value. The USART repeats the character the number of times specified in CTRLC.MAXITER.

When the USART repetition number reaches the programmed value in CTRLC.MAXITER, the STATUS.ITER bit is set and the internal iteration counter is reset. If the repetition of the character is acknowledged by the receiver before the maximum iteration is reached, the repetitions are stopped and the iteration counter is cleared.

#### Disable Successive Receive NACK

The receiver can limit the number of successive NACKs sent back to the remote transmitter. This is programmed by setting the Disable Successive NACK bit (CTRLC.DSNACK). The maximum number of NACKs transmitted is programmed in the CTRLC.MAXITER field. As soon as the maximum is reached, the character is considered as correct, an acknowledge is sent on the line, the STATUS.ITER bit is set and the internal iteration counter is reset.

#### Protocol T=1

To transmit data in ISO7816, T=1 protocol mode, the USART is configured in CTRLA.CMODE =0 communication mode with one or two stop bits. After the stop bits are sent, the transmitter does not drive the I/O line.

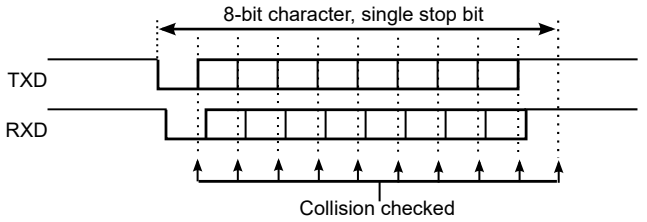
Parity is generated when transmitting and checked when receiving. Parity error detection sets the STATUS.PERR bit, and the erroneous character is written to the receive FIFO. When using T=1 protocol, the receiver does not signal errors on the I/O line and the transmitter does not retransmit.

### 34.6.3.8. Collision Detection

When the receiver and transmitter are connected either through pin configuration or externally, transmit collision can be detected after selecting the Collision Detection Enable bit in the CTRLB register (CTRLB.COLDEN=1). To detect collision, the receiver and transmitter must be enabled (CTRLB.RXEN=1 and CTRLB.TXEN=1).

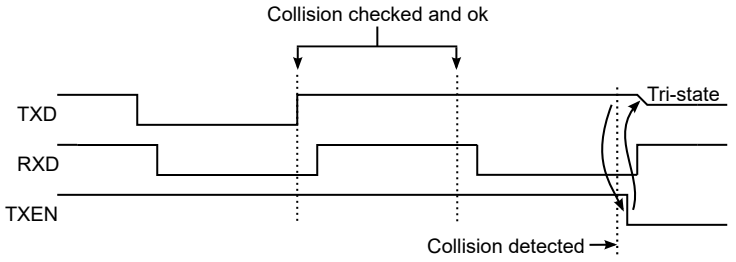
Collision detection is performed for each bit transmitted by comparing the received value with the transmit value, as shown in the figure below. While the transmitter is idle (no transmission in progress), characters can be received on RxD without triggering a collision.

**Figure 34-20.** Collision Checking




The next figure shows the conditions for a collision detection. In this case, the Start bit and the first Data bit are received with the same value as transmitted. The second received Data bit is found to be different than the transmitted bit at the detection point, which indicates a collision.

**Figure 34-21.** Collision Detected





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When a collision is detected, the USART follows this sequence:

1. Abort the current transfer.
2. Flush the transmit buffer.
3. Disable transmitter (CTRLB.TXEN=0)
   - This is done after a synchronization delay. The CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) will be set until this is complete.
   - After disabling, the TxD pin will be tri-stated.
4. Set the Collision Detected bit (STATUS.COLL) along with the Error Interrupt Flag (INTFLAG.ERROR).
5. Set the Transmit Complete Interrupt Flag (INTFLAG.TXC), since the transmit buffer no longer contains data.

After a collision, software must manually enable the transmitter again before continuing, after assuring that the CTRLB Synchronization Busy bit (SYNCBUSY.CTRLB) is not set.

#### 34.6.3.9. Loop-Back Mode

For Loop-Back mode, configure the Receive Data Pinout (CTRLA.RXPO) and Transmit Data Pinout (CTRLA.TXPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

#### 34.6.3.10. Start-of-Frame Detection

The USART start-of-frame detector can wake-up the CPU when it detects a Start bit. In Standby Sleep mode, the internal fast start-up oscillator must be selected as the GCLK_SERCOMx_CORE source, otherwise the frame data from the initial Start-of-Frame detection event may get corrupted and any succeeding frames until the oscillator clock has started up and stable due to baud rate mismatch issues.

When a 1-to-0 transition is detected on RxD, the 8 MHz Internal Oscillator is powered up and the USART clock is enabled. After start-up, the rest of the data frame can be received, provided that the baud rate is slow enough in relation to the fast start-up internal oscillator start-up time. Refer to the *Electrical Characteristics* chapters for details. The start-up time of this oscillator varies with supply voltage and temperature.

The USART start-of-frame detection works both in Asynchronous and Synchronous modes. It is enabled by writing '1' to the Start of Frame Detection Enable bit in the Control B register (CTRLB.SFDE).

If the Receive Start Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.RXS) is set, the Receive Start interrupt is generated immediately when a start is detected.

When using start-of-frame detection without the Receive Start interrupt, start detection will force the 8 MHz internal oscillator and USART clock active while the frame is being received. In this case, the CPU will not wake up until the receive complete interrupt is generated.

#### 34.6.3.11. Sample Adjustment

In Asynchronous mode (CTRLA.CMODE=0), three samples in the middle are used to determine the value based on majority voting. The three samples used for voting can be selected using the Sample Adjustment bit field in Control A register (CTRLA.SAMPA). When CTRLA.SAMPA=0, samples 7-8-9 are used for 16x oversampling, and samples 3-4-5 are used for 8x oversampling.

#### 34.6.3.12. 32-bit Extension

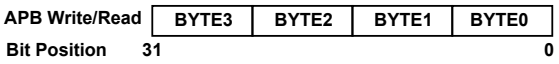
For better system bus utilization, 32-bit data receive and transmit can be enabled separately by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B). When enabled, writes and/or reads to the DATA register are 32 bit in size.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

If frames are not multiples of 4 Bytes, the length counter (LENGTH.LEN) and length enable (LENGTH.LENEN) must be configured before data transfer begins, LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

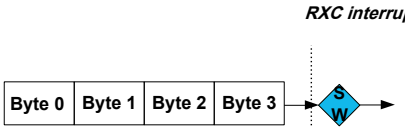
The figure below shows the order of transmit and receive when using 32-bit extension. Bytes are transmitted or received, and stored in order from 0 to 3. Only 8-bit and smaller character sizes are supported. If the character size is less than 8 bits, characters will still be 8-bit aligned within the 32-bit APB write or read. The unused bits within each byte will be zero for received data and unused for transmit data.

**Figure 34-22.** 32-bit Extension Ordering




A receive transaction using 32-bit extension is in the next figure. The Receive Complete flag (INTFLAG.RXC) is raised every four received Bytes. For transmit transactions, the Data Register Empty flag (INTFLAG.DRE) is raised instead of INTFLAG.RXC.

**Figure 34-23.** 32-bit Extension Receive Operation




### Data Length Configuration

When the Data Length Enable bit field in the Length register (LENGTH.LENEN) is written to 0x1 or 0x2, the Data Length bit (LENGTH.LEN) determines the number of characters to be transmitted or received from 1 to 255.

**Note:** There is one internal length counter that can be used for either transmit (LENGTH.LENEN=0x1) or receive (LENGTH.LENEN=0x2), but not for both simultaneously.

The LENGTH register must be written before the frame begins. If LENGTH.LEN is not a multiple of 4 Bytes, the final INTFLAG.RXC/DRE interrupt will be raised when the last byte is received/sent. The internal length counter is reset when LENGTH.LEN is reached or when LENGTH.LENEN is written to 0x0.

Writing the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 Bytes, the remainder may be lost. Attempting to use the length counter for transmit and receive at the same time will produce unpredictable results.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.6.4. DMA, Interrupts and Events

**Table 34-4.** Module Request for SERCOM USART

| Condition | Request | | |
|---|---|---|---|
| | DMA | Interrupt | Event |
| Data Register Empty (DRE) | Yes<br>(request cleared when data is written) | Yes | NA |
| Receive Complete (RXC) | Yes<br>(request cleared when data is read) | Yes | |
| Transmit Complete (TXC) | NA | Yes | |
| Receive Start (RXS) | NA | Yes | |
| Clear to Send Input Change (CTSIC) | NA | Yes | |
| Receive Break (RXBRK) | NA | Yes | |
| Error (ERROR) | NA | Yes | |

#### 34.6.4.1. DMA Operation

The USART generates the following DMA requests:

- Data received (RX): The request is set when data is available in the two-level RX buffer. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

#### 34.6.4.2. Interrupts

The USART has the following interrupt sources. These are asynchronous interrupts, and can wake-up the device from any Sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- Receive Start (RXS)
- Clear to Send Input Change (CTSIC)
- Received Break (RXBRK)
- Error (ERROR)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the Interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the Interrupt flag is cleared, the interrupt is disabled, or the USART is reset. For details on clearing Interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller

#### 34.6.4.3. Events

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.6.5. Sleep Mode Operation

The behavior in Sleep mode is depending on the clock source and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Internal clocking, CTRLA.RUNSTDBY=1: GCLK_SERCOMx_CORE can be enabled in all Sleep modes. Any interrupt can wake-up the device.
- External clocking, CTRLA.RUNSTDBY=1: The Receive Complete interrupt(s) can wake-up the device.
- Internal clocking, CTRLA.RUNSTDBY=0: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Complete interrupt(s) can wake-up the device.
- External clocking, CTRLA.RUNSTDBY=0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

### 34.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- Transmitter Enable bit in the Control B register (CTRLB.TXEN)

**Note:** CTRLB.RXEN is write-synchronized somewhat differently. See also CTRLB for details.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 34.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | RUNSTDBY | | | | MODE[2:0] | | ENABLE | SWRST |
| | | 15:8 | | SAMPR[2:0] | | | | RXINV | TXINV | IBON |
| | | 23:16 | | SAMPA[1:0] | | RXPO[1:0] | | | | TXPO[1:0] |
| | | 31:24 | | | DORD | CPOL | CMODE | | FORM[3:0] | |
| 0x04 | CTRLB | 7:0 | | | SBMODE | | | | CHSIZE[2:0] | |
| | | 15:8 | | | | PMODE | | ENC | SFDE | COLDEN |
| | | 23:16 | | | | | | | RXEN | TXEN |
| | | 31:24 | | | | | | | LINCMD[1:0] | |
| 0x08 | CTRLC | 7:0 | | | | | | | GTIME[2:0] | |
| | | 15:8 | | | | | | HDRDLY[1:0] | | BRKLEN[1:0] |
| | | 23:16 | | | | MAXITER[2:0] | | | DSNACK | INACK |
| | | 31:24 | | | | | | | DATA32B[1:0] | |
| 0x0C | BAUD | 7:0 | | | | BAUD[7:0] | | | | |
| | | 15:8 | | | | BAUD[15:8] | | | | |
| 0x0E | RXPL | 7:0 | | | | RXPL[7:0] | | | | |
| 0x0F ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | ERROR | | RXBRK | CTSIC | RXS | RXC | TXC | DRE |
| 0x15 | Reserved | | | | | | | | | |
| 0x16 | INTENSET | 7:0 | ERROR | | RXBRK | CTSIC | RXS | RXC | TXC | DRE |
| 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTFLAG | 7:0 | ERROR | | RXBRK | CTSIC | RXS | RXC | TXC | DRE |
| 0x19 | Reserved | | | | | | | | | |
| 0x1A | STATUS | 7:0 | ITER | TXE | COLL | ISF | CTS | BUFOVF | FERR | PERR |
| | | 15:8 | | | | | | | | |
| 0x1C | SYNCBUSY | 7:0 | | | | LENGTH | RXERRCNT | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | RXERRCNT | 7:0 | | | | RXERRCNT[7:0] | | | | |
| 0x21 | Reserved | | | | | | | | | |
| 0x22 | LENGTH | 7:0 | | | | LEN[7:0] | | | | |
| | | 15:8 | | | | | | | LENEN[1:0] | |
| 0x24 ... 0x27 | Reserved | | | | | | | | | |
| 0x28 | DATA | 7:0 | | | | DATA[7:0] | | | | |
| | | 15:8 | | | | DATA[15:8] | | | | |
| | | 23:16 | | | | DATA[23:16] | | | | |
| | | 31:24 | | | | DATA[31:24] | | | | |
| 0x2C ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DBGCTRL | 7:0 | | | | | | | | DBGSTOP |

## 34.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | DORD | CPOL | CMODE | FORM[3:0] | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SAMPA[1:0] | | RXPO[1:0] | | | | TXPO[1:0] | |
| Access | R/W | R/W | R/W | R/W | | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SAMPR[2:0] | | | | | RXINV | TXINV | IBON |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 30 – DORD** Data Order
This bit selects the data order when a character is shifted out from the Data register.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | MSB is transmitted first. |
| 1 | LSB is transmitted first. |

**Bit 29 – CPOL** Clock Polarity
This bit selects the relationship between data output change and data input sampling in synchronous mode.
This bit is not synchronized.

| CPOL | TxD Change | RxD Sample |
|---|---|---|
| 0x0 | Rising XCK edge | Falling XCK edge |
| 0x1 | Falling XCK edge | Rising XCK edge |

**Bit 28 – CMODE** Communication Mode
This bit selects asynchronous or synchronous communication.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Asynchronous communication. |
| 1 | Synchronous communication. |

**Bits 27:24 – FORM[3:0]** Frame Format
These bits define the frame format.
These bits are not synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| FORM[3:0] | Description |
|---|---|
| 0x0 | USART frame |
| 0x1 | USART frame with parity |
| 0x2 | LIN Host - Break and sync generation. See LIN Command (CTRLB.LINCMD). |
| 0x3 | Reserved |
| 0x4 | Auto-baud (LIN Client) - break detection and auto-baud. |
| 0x5 | Auto-baud - break detection and auto-baud with parity |
| 0x6 | Reserved |
| 0x7 | ISO 7816 |
| 0x8-0xF | Reserved |

**Bits 23:22 – SAMPA[1:0]** Sample Adjustment
These bits define the sample adjustment.
These bits are not synchronized.

| SAMPA[1:0] | 16x Over-sampling (CTRLA.SAMPR=0 or 1) | 8x Over-sampling (CTRLA.SAMPR=2 or 3) |
|---|---|---|
| 0x0 | 7-8-9 | 3-4-5 |
| 0x1 | 9-10-11 | 4-5-6 |
| 0x2 | 11-12-13 | 5-6-7 |
| 0x3 | 13-14-15 | 6-7-8 |

**Bits 21:20 – RXPO[1:0]** Receive Data Pinout
These bits define the receive data (RxD) pin configuration.
These bits are not synchronized.

| RXPO[1:0] | Name | Description |
|---|---|---|
| 0x0 | PAD[0] | SERCOM PAD[0] is used for data reception |
| 0x1 | PAD[1] | SERCOM PAD[1] is used for data reception |
| 0x2 | PAD[2] | SERCOM PAD[2] is used for data reception |
| 0x3 | PAD[3] | SERCOM PAD[3] is used for data reception |

**Bits 17:16 – TXPO[1:0]** Transmit Data Pinout
These bits define the transmit data (TxD) and XCK pin configurations.
This bit is not synchronized.

| TXPO | TxD Pin Location | XCK Pin Location (When Applicable) | RTS/TE | CTS |
|---|---|---|---|---|
| 0x0 | SERCOM PAD[0] | SERCOM PAD[1] | N/A | N/A |
| 0x1 | Reserved | | | |
| 0x2 | SERCOM PAD[0] | N/A | SERCOM PAD[2] | SERCOM PAD[3] |
| 0x3 | SERCOM_PAD[0] | SERCOM_PAD[1] | SERCOM_PAD[2] | N/A |

**Bits 15:13 – SAMPR[2:0]** Sample Rate
These bits select the sample rate.
These bits are not synchronized.

| SAMPR[2:0] | Description |
|---|---|
| 0x0 | 16x over-sampling using arithmetic baud rate generation. |
| 0x1 | 16x over-sampling using fractional baud rate generation. |
| 0x2 | 8x over-sampling using arithmetic baud rate generation. |
| 0x3 | 8x over-sampling using fractional baud rate generation. |
| 0x4 | 3x over-sampling using arithmetic baud rate generation. |
| 0x5-0x7 | Reserved |

**Bit 10 – RXINV** Receive Data Invert
This bit controls whether the receive data (RxD) is inverted or not.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. Start, parity, and stop bits are unchanged. When enabled, parity is calculated on the inverted data.
2. This field is only valid when using ISO7816 T = 0 mode (CTRLA.MODE = 0x7).

| Value | Description |
|---|---|
| 0 | RxD is not inverted. |
| 1 | RxD is inverted. |

**Bit 9 – TXINV** Transmit Data Invert

This bit controls whether the transmit data (TxD) is inverted or not.

**Notes:**

1. Start, parity, and stop bits are unchanged. When enabled, parity is calculated on the inverted data.
2. This field is only valid when using ISO7816 T = 0 mode (CTRLA.MODE = 0x7).

| Value | Description |
|---|---|
| 0 | TxD is not inverted. |
| 1 | TxD is inverted. |

**Bit 8 – IBON** Immediate Buffer Overflow Notification

This bit controls when the buffer overflow status bit (STATUS.BUFOVF) is asserted when a buffer overflow occurs.

| Value | Description |
|---|---|
| 0 | STATUS.BUFOVF is asserted when it occurs in the data stream. |
| 1 | STATUS.BUFOVF is asserted immediately upon buffer overflow. |

**Bit 7 – RUNSTDBY** Run In Standby

This bit defines the functionality in Standby sleep mode.
This bit is not synchronized.

| RUNSTDBY | External Clock | Internal Clock |
|---|---|---|
| 0x0 | External clock is disconnected when ongoing transfer is finished. All reception is dropped. | Generic clock is disabled when ongoing transfer is finished. The device will not wake up on Transfer Complete interrupt unless the appropriate ONDEMAND bits are set in the clocking chain. |
| 0x1 | Wake on Receive Complete interrupt. | Generic clock is enabled in all sleep modes. Any interrupt can wake up the device. |

**Bits 4:2 – MODE[2:0]** Operating Mode

These bits select the USART serial communication interface of the SERCOM.
These bits are not synchronized.

| Value | Description |
|---|---|
| 0x0 | USART with external clock |
| 0x1 | USART with internal clock |

**Bit 1 – ENABLE** Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled or being disabled. |
| 1 | The peripheral is enabled or being enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – SWRST** Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | LINCMD[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RXEN | TXEN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | PMODE | | | ENC | SFDE | COLDEN |
| Access | | | R/W | | | R/W | R/W | R/W |
| Reset | | | 0 | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | SBMODE | | | | CHSIZE[2:0] | | |
| Access | | R/W | | | | R/W | R/W | R/W |
| Reset | | 0 | | | | 0 | 0 | 0 |

**Bits 25:24 – LINCMD[1:0]** LIN Command

These bits define the LIN header transmission control. This field is only valid in LIN Host mode (CTRLA.FORM = LIN Host).
These are strobe bits and will always read back as zero.
These bits are not enable-protected.

| Value | Description |
|---|---|
| 0x0 | Normal USART transmission. |
| 0x1 | Break field is transmitted when DATA is written. |
| 0x2 | Break, sync, and identifier are automatically transmitted when DATA is written with the identifier. |
| 0x3 | Reserved |

**Bit 17 – RXEN** Receiver Enable

Writing '0' to this bit will disable the USART receiver. Disabling the receiver will flush the receive buffer and clear the FERR, PERR, and BUFOVF bits in the STATUS register.
Writing '1' to CTRLB.RXEN when the USART is disabled will set CTRLB.RXEN immediately. When the USART is enabled, CTRLB.RXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled, CTRLB.RXEN will read back as '1'.
Writing '1' to CTRLB.RXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The receiver is disabled or being enabled. |
| 1 | The receiver is enabled or will be enabled when the USART is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 16 – TXEN** Transmitter Enable

Writing '0' to this bit will disable the USART transmitter. Disabling the transmitter will not become effective until ongoing and pending transmissions are completed.
Writing '1' to CTRLB.TXEN when the USART is disabled will set CTRLB.TXEN immediately. When the USART is enabled, CTRLB.TXEN will be cleared, and SYNCBUSY.CTRLB will be set and remain set until the transmitter is enabled. When the transmitter is enabled, CTRLB.TXEN will read back as '1'.
Writing '1' to CTRLB.TXEN when the USART is enabled will set SYNCBUSY.CTRLB, which will remain set until the transmitter is enabled, and CTRLB.TXEN will read back as '1'.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The transmitter is disabled or being enabled. |
| 1 | The transmitter is enabled or will be enabled when the USART is enabled. |

**Bit 13 – PMODE** Parity Mode

This bit selects the type of parity used when parity is enabled (CTRLA.FORM is 0x1 or 0x5). The transmitter will automatically generate and send the parity of the transmitted data bits within each frame. The receiver will generate a parity value for the incoming data and parity bit, compare it to the parity mode and, if a mismatch is detected, STATUS.PERR will be set.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Even parity. |
| 1 | Odd parity. |

**Bit 10 – ENC** Encoding Format

This bit selects the data encoding format.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Data is not encoded. |
| 1 | Data is IrDA encoded. |

**Bit 9 – SFDE** Start of Frame Detection Enable

This bit controls whether the start-of-frame detector will wake up the device when a start bit is detected on the RxD line.
This bit is not synchronized.

| SFDE | INTENSET.RXS | INTENSET.RXC | Description |
|---|---|---|---|
| 0 | X | X | Start-of-frame detection disabled. |
| 1 | 0 | 0 | Reserved |
| 1 | 0 | 1 | Start-of-frame detection enabled. RXC wakes up the device from all sleep modes. |
| 1 | 1 | 0 | Start-of-frame detection enabled. RXS wakes up the device from all sleep modes. |
| 1 | 1 | 1 | Start-of-frame detection enabled. Both RXC and RXS wake up the device from all sleep modes. |

**Bit 8 – COLDEN** Collision Detection Enable

This bit enables collision detection.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Collision detection is not enabled. |
| 1 | Collision detection is enabled. |

**Bit 6 – SBMODE** Stop Bit Mode

This bit selects the number of stop bits transmitted.
This bit is not synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | One stop bit. |
| 1 | Two stop bits. |

**Bits 2:0 – CHSIZE[2:0]** Character Size
These bits select the number of bits in a character.
These bits are not synchronized.

| CHSIZE[2:0] | Description |
|---|---|
| 0x0 | 8 bits |
| 0x1 | 9 bits |
| 0x2-0x4 | Reserved |
| 0x5 | 5 bits |
| 0x6 | 6 bits |
| 0x7 | 7 bits |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 34.8.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DATA32B[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | MAXITER[2:0] | | | | | DSNACK | INACK |
| Access | | R/W | R/W | R/W | | | R/W | R/W |
| Reset | | 0 | 0 | 0 | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | HDRDLY[1:0] | | BRKLEN[1:0] | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | GTIME[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 25:24 – DATA32B[1:0]** Data 32 Bit
These bits configure 32-bit Extension for read and write transactions to the DATA register.
When disabled, access is according to CTRLB.CHSIZE.

| Value | Description |
|---|---|
| 0x0 | DATA reads (for received data) and writes (for transmit data) according to CTRLB.CHSIZE. |
| 0x1 | DATA reads according to CTRLB.CHSIZE.<br>DATA writes using 32-bit Extension. |
| 0x2 | DATA reads using 32-bit Extension.<br>DATA writes according to CTRLB.CHSIZE. |
| 0x3 | DATA reads and writes using 32-bit Extension. |

**Bits 22:20 – MAXITER[2:0]** Maximum Iterations
These bits define the maximum number of retransmit iterations.
These bits also define the successive NACKs sent to the remote transmitter when CTRLC.DSNACK is set.
This field is only valid when using ISO7816 T = 0 mode (CTRLA.MODE = 0x7 and CTRLA.CMODE=0).

**Bit 17 – DSNACK** Disable Successive Not Acknowledge
This bit controls how many times NACK will be sent on parity error reception.
This bit is only valid in ISO7816 T = 0 mode and when CTRLC.INACK = 0.

| Value | Description |
|---|---|
| 0 | NACK is sent on the ISO line for every parity error received. |
| 1 | Successive parity errors are counted up to the value specified in CTRLC.MAXITER. These parity errors generate a NACK on the ISO line. As soon as this value is reached, no additional NACK is sent on the ISO line. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 16 – INACK** Inhibit Not Acknowledge

This bit controls whether a NACK is transmitted when a parity error is received.
This bit is only valid in ISO7816 T = 0 mode.

| Value | Description |
|---|---|
| 0 | NACK is transmitted when a parity error is received. |
| 1 | NACK is not transmitted when a parity error is received. |

**Bits 11:10 – HDRDLY[1:0]** LIN Host Header Delay

These bits define the delay between break and sync transmission in addition to the delay between the sync and identifier (ID) fields when in LIN Host mode (CTRLA.FORM = 0x2).
This field is only valid when using the LIN header command (CTRLB.LINCMD = 0x2).
**Note:** A correct setting for this bit field must be selected in order to follow the LIN specification.

| Value | Description |
|---|---|
| 0x0 | Delay between break and sync transmission is 1 bit time.<br>Delay between sync and ID transmission is 1 bit time. |
| 0x1 | Delay between break and sync transmission is 4 bit time.<br>Delay between sync and ID transmission is 4 bit time. |
| 0x2 | Delay between break and sync transmission is 8 bit time.<br>Delay between sync and ID transmission is 4 bit time. |
| 0x3 | Delay between break and sync transmission is 14 bit time.<br>Delay between sync and ID transmission is 4 bit time. |

**Bits 9:8 – BRKLEN[1:0]** LIN Host Break Length

These bits define the length of the break field transmitted when in LIN Host mode (CTRLA.FORM = 0x2).
**Note:** A correct setting for this bit field must be selected to follow the LIN specification.

| Value | Description |
|---|---|
| 0x0 | Break field transmission is 13 bit times |
| 0x1 | Break field transmission is 17 bit times |
| 0x2 | Break field transmission is 21 bit times |
| 0x3 | Break field transmission is 26 bit times |

**Bits 2:0 – GTIME[2:0]** Guard Time

These bits define the guard time when using RS485 mode (CTRLA.FORM = 0x0 or CTRLA.FORM = 0x1, and CTRLA.TXPO = 0x3) or ISO7816 mode (CTRLA.FORM = 0x7).
For RS485 mode, the guard time is programmable from 0-7 bit times and defines the time that the transmit enable pin (TE) remains high after the last stop bit is transmitted and there is no remaining data to be transmitted.
For ISO7816 T = 0 mode, the guard time is programmable from 2-9 bit times and defines the guard time between each transmitted byte.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.4. Baud

**Name:** BAUD
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** Enable-Protected, PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – BAUD[15:0]** Baud Value

Arithmetic Baud Rate Generation (`CTRLA.SAMPR[0]=0`):
These bits control the clock generation, as described in the SERCOM Baud Rate section.
If Fractional Baud Rate Generation (`CTRLA.SAMPR[0]=1`) bit positions 15 to 13 are replaced by FP[2:0] Fractional Part:

- **Bits 15:13 - FP[2:0]: Fractional Part**
  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.
- **Bits 12:0 - BAUD[12:0]: Baud Value**
  These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator* section.

**Related Links**
Clock Generation – Baud-Rate Generator
Asynchronous Arithmetic Mode BAUD Value Selection


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.5. Receive Pulse Length Register

**Name:** RXPL
**Offset:** 0x0E
**Reset:** 0x00
**Property:** Enable-Protected, PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXPL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – RXPL[7:0]** Receive Pulse Length
When the encoding format is set to IrDA (CTRLB.ENC=1), these bits control the minimum pulse length that is required for a pulse to be accepted by the IrDA receiver with regards to the serial engine clock period $SE_{per}$.
$PULSE \geq (\mathrm{RXPL} + 2) \cdot SE_{per}$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.6. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | RXBRK | CTSIC | RXS | RXC | TXC | DRE |
| Access | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 5 – RXBRK** Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Break Interrupt Enable bit, which disables the Receive Break interrupt.

| Value | Description |
|---|---|
| 0 | Receive Break interrupt is disabled. |
| 1 | Receive Break interrupt is enabled. |

**Bit 4 – CTSIC** Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Clear To Send Input Change Interrupt Enable bit, which disables the Clear To Send Input Change interrupt.

| Value | Description |
|---|---|
| 0 | Clear To Send Input Change interrupt is disabled. |
| 1 | Clear To Send Input Change interrupt is enabled. |

**Bit 3 – RXS** Receive Start Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Start Interrupt Enable bit, which disables the Receive Start interrupt.

| Value | Description |
|---|---|
| 0 | Receive Start interrupt is disabled. |
| 1 | Receive Start interrupt is enabled. |

**Bit 2 – RXC** Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

| Value | Description |
|---|---|
| 0 | Receive Complete interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Receive Complete interrupt is enabled. |

**Bit 1 – TXC** Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

| Value | Description |
|---|---|
| 0 | Transmit Complete interrupt is disabled. |
| 1 | Transmit Complete interrupt is enabled. |

**Bit 0 – DRE** Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

| Value | Description |
|---|---|
| 0 | Data Register Empty interrupt is disabled. |
| 1 | Data Register Empty interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.7. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x16
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | RXBRK | CTSIC | RXS | RXC | TXC | DRE |
| Access | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 5 – RXBRK** Receive Break Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Break Interrupt Enable bit, which enables the Receive Break interrupt.

| Value | Description |
|---|---|
| 0 | Receive Break interrupt is disabled. |
| 1 | Receive Break interrupt is enabled. |

**Bit 4 – CTSIC** Clear to Send Input Change Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Clear To Send Input Change Interrupt Enable bit, which enables the Clear To Send Input Change interrupt.

| Value | Description |
|---|---|
| 0 | Clear To Send Input Change interrupt is disabled. |
| 1 | Clear To Send Input Change interrupt is enabled. |

**Bit 3 – RXS** Receive Start Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Start Interrupt Enable bit, which enables the Receive Start interrupt.

| Value | Description |
|---|---|
| 0 | Receive Start interrupt is disabled. |
| 1 | Receive Start interrupt is enabled. |

**Bit 2 – RXC** Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

| Value | Description |
|---|---|
| 0 | Receive Complete interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Receive Complete interrupt is enabled. |

**Bit 1 – TXC** Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

| Value | Description |
|---|---|
| 0 | Transmit Complete interrupt is disabled. |
| 1 | Transmit Complete interrupt is enabled. |

**Bit 0 – DRE** Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

| Value | Description |
|---|---|
| 0 | Data Register Empty interrupt is disabled. |
| 1 | Data Register Empty interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.8. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | RXBRK | CTSIC | RXS | RXC | TXC | DRE |
| Access | R/W | | R/W | R/W | R/W | R | R/W | R |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. Errors that will set this flag are COLL, ISF, BUFOVF, FERR, and PERR.Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 5 – RXBRK** Receive Break
This flag is cleared by writing '1' to it.
This flag is set when auto-baud is enabled (CTRLA.FORM) and a break character is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 4 – CTSIC** Clear to Send Input Change
This flag is cleared by writing a '1' to it.
This flag is set when a change is detected on the CTS pin.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 3 – RXS** Receive Start
This flag is cleared by writing '1' to it.
This flag is set when a Start condition is detected on the RxD line and start-of-frame detection is enabled (CTRLB.SFDE is '1').
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Start Interrupt flag.

**Bit 2 – RXC** Receive Complete
This flag is cleared by reading the Data register (DATA) or by disabling the receiver.
This flag is set when there are unread data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

**Bit 1 – TXC** Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
This flag is set when the entire frame in the Transmit Shift register has been shifted out and there are no new data in DATA.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 0 – DRE** Data Register Empty
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready to be written.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.9. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ITER | TXE | COLL | ISF | CTS | BUFOVF | FERR | PERR |
| Access | R/W | R/W | R/W | R/W | R | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ITER** Maximum Number of Repetitions Reached
This bit is set when the maximum number of NACK repetitions or retransmissions is met in ISO7816 T=0 mode.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

**Bit 6 – TXE** Transmitter Empty
When CTRLA.FORM is set to LIN Host mode, this bit is set when any ongoing transmission is complete and TxDATA is empty.
When CTRLA.FORM is not set to LIN Host mode, this bit will always read back as zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

**Bit 5 – COLL** Collision Detected
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when collision detection is enabled (CTRLB.COLDEN) and a collision is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

**Bit 4 – ISF** Inconsistent Sync Field
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when the frame format is set to auto-baud (CTRLA.FORM) and a sync field not equal to 0x55 is received.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

**Bit 3 – CTS** Clear to Send
This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

**Bit 2 – BUFOVF** Buffer Overflow
Reading this bit before reading the Data register will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit is set when a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full, there is a new character waiting in the receive shift register and a new start bit is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

**Bit 1 – FERR** Frame Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if the received character had a frame error, i.e., when the first stop bit is zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

**Bit 0 – PERR** Parity Error
Reading this bit before reading the Data register will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set if parity checking is enabled (CTRLA.FORM is 0x1, 0x5, or 0x7) and a parity error is detected.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.10. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | LENGTH | RXERRCNT | CTRLB | ENABLE | SWRST |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 4 – LENGTH** LENGTH Synchronization Busy

Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH will be set until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, an APB error is generated.

| Value | Description |
|---|---|
| 0 | LENGTH synchronization is not busy. |
| 1 | LENGTH synchronization is busy. |

**Bit 3 – RXERRCNT** Receive Error Count Synchronization Busy

The RXERRCNT register is automatically synchronized to the APB domain upon error. When returning from sleep, this bit will be raised until the new value is available to be read.

| Value | Description |
|---|---|
| 0 | RXERRCNT synchronization is not busy. |
| 1 | RXERRCNT synchronization is busy. |

**Bit 2 – CTRLB** CTRLB Synchronization Busy

Writing to the CTRLB register when the SERCOM is enabled requires synchronization. When writing to CTRLB the SYNCBUSY.CTRLB bit will be set until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB is asserted, an APB error will be generated.

| Value | Description |
|---|---|
| 0 | CTRLB synchronization is not busy. |
| 1 | CTRLB synchronization is busy. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 1 – ENABLE** SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

| Value | Description |
|---|---|
| 0 | Enable synchronization is not busy. |
| 1 | Enable synchronization is busy. |

**Bit 0 – SWRST** Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

| Value | Description |
|---|---|
| 0 | SWRST synchronization is not busy. |
| 1 | SWRST synchronization is busy. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.11. Receive Error Count

**Name:** RXERRCNT
**Offset:** 0x20
**Reset:** 0x00
**Property:** Read-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RXERRCNT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – RXERRCNT[7:0]** Receive Error Count
This register records the total number of parity errors and NACK errors combined in ISO7816 mode (CTRLA.FORM=0x7).
This register is automatically cleared on read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.12. Length

**Name:** LENGTH
**Offset:** 0x22
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | LENEN[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LEN[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 9:8 – LENEN[1:0]** Data Length Enable
In 32-bit Extension mode, this bit field configures the length counter either for transmit or receive transactions.

| Value | Description |
|---|---|
| 0x0 | Length counter disabled |
| 0x1 | Length counter enabled for transmit |
| 0x2 | Length counter enabled for receive |
| 0x3 | Reserved |

**Bits 7:0 – LEN[7:0]** Data Length
In 32-bit Extension mode, this bit field configures the data length after which the flags INTFLAG.RXC or INTFLAG.DRE are raised.

| Value | Description |
|---|---|
| 0x00 | Reserved if LENEN=0x1 or LENEN=0x2 |
| 0x01-0xFF | Data Length |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.13. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x0000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Data

Reading these bits will return the contents of the Receive Data register. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set. The status bits in STATUS should be read before reading the DATA value in order to get any corresponding error.

Writing these bits will write the Transmit Data register. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRLC.DATA32B setting.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.8.14. Debug Control

**Name:** DBGCTRL
**Offset:** 0x30
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGSTOP |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGSTOP** Debug Stop Mode
This bit controls the baud-rate generator functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The baud-rate generator continues normal operation when the CPU is halted by an external debugger. |
| 1 | The baud-rate generator is halted when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 35. SERCOM SPI – SERCOM Serial Peripheral Interface

## 35.1. Overview

The Serial Peripheral Interface (SPI) is one of the available modes in the Serial Communication Interface (SERCOM).

The SPI uses the SERCOM transmitter and receiver configured as shown in Block Diagram. Each side, host and client, depicts a separate SPI containing a Shift register, a transmit buffer and a two-level receive buffer. In addition, the SPI host uses the SERCOM baud-rate generator, while the SPI client can use the SERCOM address match logic. Labels in capital letters are synchronous to CLK_SERCOMx_APB and accessible by the CPU, while labels in lowercase letters are synchronous to the SCK clock.

**Related Links**

SERCOM – Serial Communication Interface

## 35.2. Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (MISO, MOSI, SCK, $\overline{SS}$)
- One-level transmit buffer, two-level receive buffer
- Supports all four SPI modes of operation
- Single data direction operation allows alternate function on MISO or MOSI pin
- Selectable LSB-first data transfer or MSB-first data transfer
- Can be used with DMA
- 32-bit Extension for better system bus utilization
- Host operation:
  - Serial clock speed, $f_{SCK}=1/t_{SCK}$(1)
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$
  - Optional inter-character spacing
- Client Operation:
  - Serial clock speed, $f_{SCK} = 1/t_{SSCK}$(1)
  - Optional 8-bit address match operation
  - Operation in all sleep modes
  - Wake on $\overline{SS}$ transition

1. For $t_{SCK}$ and $t_{SSCK}$ values, refer to “SPI Timing Characteristics”.

**Related Links**

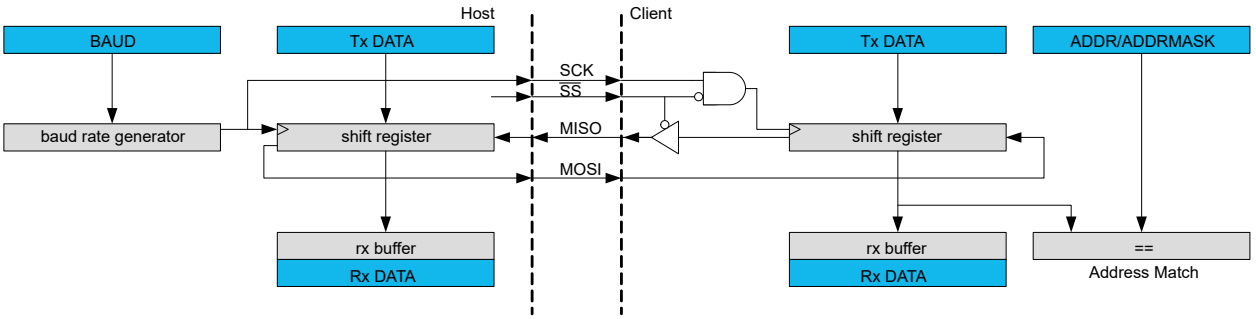
SERCOM – Serial Communication Interface
Features


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 35.3. Block Diagram

**Figure 35-1.** Full-Duplex SPI Host Client Interconnection




## 35.4. Signal Description

**Table 35-1.** SERCOM SPI Signals

| Signal Name | Type | Description |
|---|---|---|
| PAD[3:0] | Digital I/O | General SERCOM pins |

One signal can be mapped to one of several pins.

**Related Links**
I/O Multiplexing and Considerations

## 35.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 35.5.1. I/O Lines

To use the SERCOM's I/O lines, the I/O pins must be configured using the I/O Pin Controller (PORT).

When the SERCOM is configured for SPI operation, the SERCOM controls the direction and value of the I/O pins according to the table below. The PORT Control bit (PINCFGn.DRVSTR) is still effective for the SERCOM output pins.

The PORT Control bit (PINCFGn.PULLEN) is still effective on the SERCOM input pins, but is limited to the enabling or disabling of a pull down only (it is not possible to enable/disable a pull up).

If the receiver is disabled, the data input pin can be used for other purposes.

In Host mode, the Client Select line ($\overline{SS}$) is hardware controlled when the Host Client Select Enable bit in the Control B register (CTRLB.MSSEN) is '1'.

**Table 35-2.** SPI Pin Configuration

| Pin | Host SPI | Client SPI |
|---|---|---|
| MOSI | Output | Input |
| MISO | Input | Output |
| SCK | Output | Input |
| $\overline{SS}$ | Output (CTRLB.MSSEN=1) | Input |

The combined configuration of PORT, the Data In Pinout and the Data Out Pinout bit groups in the Control A register (CTRLA.DIPO and CTRLA.DOPO) define the physical position of the SPI signals as showin in the table above.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
PORT - I/O Pin Controller

### 35.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

**Related Links**
PM – Power Manager

### 35.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

A generic clock (GCLK_SERCOMx_CORE) is required to clock the SPI. This clock must be configured and enabled in the Generic Clock Controller before using the SPI.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writes to certain registers will require synchronization to the clock domains.

**Related Links**
GCLK - Generic Clock Controller
Peripheral Clock Masking
Synchronization

### 35.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

**Related Links**
DMAC – Direct Memory Access Controller

### 35.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller

### 35.5.6. Events

Not applicable.

### 35.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

### 35.5.8. Register Access Protection

Registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC).

PAC write protection is not available for the following registers:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)

Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description.

Write-protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 35.5.9. Analog Connections

Not applicable.

## 35.6. Functional Description

### 35.6.1. Principle of Operation

The SPI is a high-speed synchronous data transfer interface. It allows high-speed communication between the device and peripheral devices.

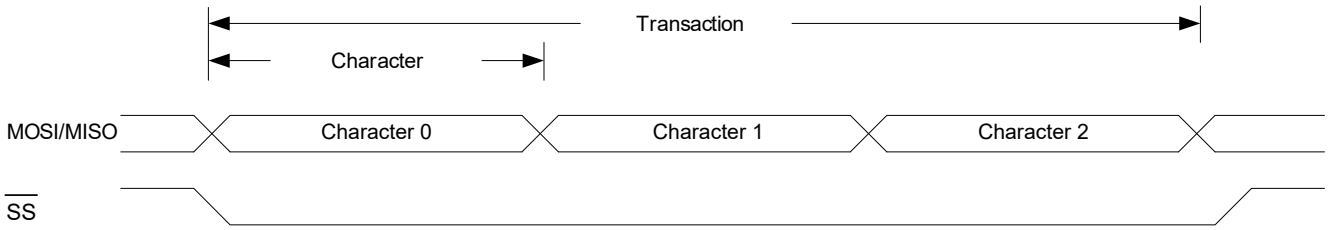
The SPI can operate as Host or Client. As Host, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Data register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the two-level receive buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format. Each transaction can contain one or more characters. The character size is configurable, and can be either 8 or 9 bits.

**Figure 35-2.** SPI Transaction Format




The SPI Host must pull the SPI select line ($\overline{SS}$) of the desired Client low to initiate a transaction. The Host and Client prepare data to send via their respective Shift registers, and the Host generates the serial clock on the SCK line.

Data are always shifted from Host to Client on the Host Output Client Input line (MOSI); data is shifted from Client to Host on the Host Input Client Output line (MISO).

Each time character is shifted out from the Host, a character will be shifted out from the Client simultaneously. To signal the end of a transaction, the Host will pull the $\overline{SS}$ line high

### 35.6.2. Basic Operation

#### 35.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRL.ENABLE=0):


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB), except Receiver Enable (CTRLB.RXEN)
- Baud register (BAUD)
- Address register (ADDR)

When the SPI is enabled or is being enabled (CTRLA.ENABLE=1), any writing to these registers will be discarded.

When the SPI is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the Enable-Protection property in the register description.

Initialize the SPI by following these steps:

1. Select SPI mode in host/client operation in the Operating Mode bit group in the CTRLA register (CTRLA.MODE= 0x2 or 0x3 ).
2. Select Transfer mode for the Clock Polarity bit and the Clock Phase bit in the CTRLA register (CTRLA.CPOL and CTRLA.CPHA) if desired.
3. Select the Frame Format value in the CTRLA register (CTRLA.FORM).
4. Configure the Data In Pinout field in the Control A register (CTRLA.DIPO) for SERCOM pads of the receiver.
5. Configure the Data Out Pinout bit group in the Control A register (CTRLA.DOPO) for SERCOM pads of the transmitter.
6. Select the Character Size value in the CTRLB register (CTRLB.CHSIZE).
7. Write the Data Order bit in the CTRLA register (CTRLA.DORD) for data direction.
8. If the SPI is used in Host mode:
   a. Select the desired baud rate by writing to the Baud register (BAUD).
   b. If Hardware $\overline{SS}$ control is required, write '1' to the Host SPI Select Enable bit in CTRLB register (CTRLB.MSSEN).
9. Enable the receiver by writing the Receiver Enable bit in the CTRLB register (CTRLB.RXEN=1).

#### 35.6.2.2. Enabling, Disabling, and Resetting

This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

Refer to the CTRLA register description for details.

#### 35.6.2.3. Clock Generation

In the SPI host operation (CTRLA.MODE=0x3), the serial clock (SCK) is generated internally by the SERCOM Baud Rate Generator (BRG).

In SPI mode, the BRG is set to Synchronous mode. The 8-bit Baud register (BAUD) value is used for generating SCK and clocking the Shift register. Refer to *Clock Generation – Baud-Rate Generator* for more details.

In SPI client operation (CTRLA.MODE is 0x2), the clock is provided by an external host on the SCK pin. This clock is used to clock the SPI Shift register.

**Related Links**

Clock Generation – Baud-Rate Generator


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 35.6.2.4. Data Register

The SPI Transmit Data register (TxDATA) and SPI Receive Data register (RxDATA) share the same I/O address, referred to as the SPI Data register (DATA). Writing DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

#### 35.6.2.5. SPI Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI Data Transfer modes are shown in SPI Transfer Modes (Table) and SPI Transfer Modes (Figure).

SCK phase is configured by the Clock Phase bit in the CTRLA register (CTRLA.CPHA). SCK polarity is programmed by the Clock Polarity bit in the CTRLA register (CTRLA.CPOL). Data bits are shifted out and latched in on opposite edges of the SCK signal. This ensures sufficient time for the data signals to stabilize.

**Table 35-3.** SPI Transfer Modes

| Mode | CPOL | CPHA | Leading Edge | Trailing Edge |
|---|---|---|---|---|
| 0 | 0 | 0 | Rising, sample | Falling, setup |
| 1 | 0 | 1 | Rising, setup | Falling, sample |
| 2 | 1 | 0 | Falling, sample | Rising, setup |
| 3 | 1 | 1 | Falling, setup | Rising, sample |

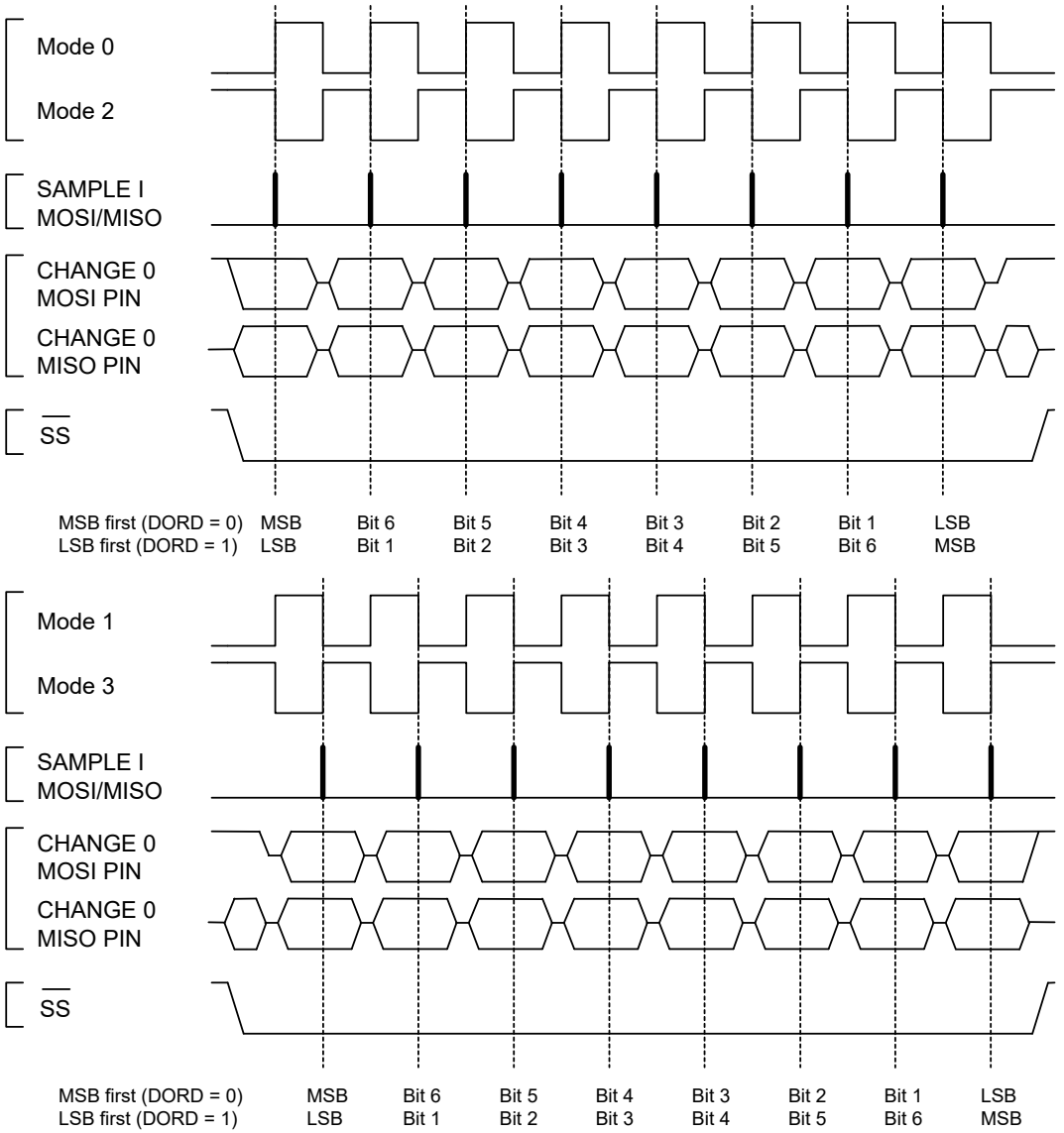
**Note:**
Leading edge is the first clock edge in a clock cycle.

Trailing edge is the second clock edge in a clock cycle.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Figure 35-3.** SPI Transfer Modes




### 35.6.2.6. Transferring Data

#### 35.6.2.6.1. Host

In Host mode (CTRLA.MODE=0x3), when Host SPI Select Enable (CTRLB.MSSEN) is '1', hardware will control the $\overline{SS}$ line.

When Host SPI Select Enable (CTRLB.MSSEN) is '0', the $\overline{SS}$ line must be configured as an output. $\overline{SS}$ can be assigned to any general purpose I/O pin. When the SPI is ready for a data transaction, software must pull the $\overline{SS}$ line low.

When writing a character to the Data register (DATA), the character will be transferred to the Shift register. Once the content of TxDATA has been transferred to the Shift register, the Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) will be set. And a new character can be written to DATA.

Each time one character is shifted out from the Host, another character will be shifted in from the Client simultaneously. If the receiver is enabled (CTRLA.RXEN=1), the contents of the Shift register will be transferred to the two-level receive buffer. The transfer takes place in the same clock cycle as


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

the last data bit is shifted in. And the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set. The received data can be retrieved by reading DATA.

When the last character has been transmitted and there is no valid data in DATA, the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set. When the transaction is finished, the Host must pull the $\overline{SS}$ line high to notify the Client. If Host SPI Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

#### 35.6.2.6.2. Client

In Client mode (CTRLA.MODE=0x2), the SPI interface will remain inactive with the MISO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of DATA at any time as long as the Data Register Empty flag in the Interrupt Status and Clear register (INTFLAG.DRE) is set.

When $\overline{SS}$ is pulled low and SCK is running, the client will sample and shift out data according to the Transaction mode set. When the content of TxDATA has been loaded into the Shift register, INTFLAG.DRE will be set, and new data can be written to DATA.

Similar to the host, the client will receive one character for each character transmitted. A character will be transferred into the two-level receive buffer within the same clock cycle its last data bit is received. The received character can be retrieved from DATA when the Receive Complete interrupt flag (INTFLAG.RXC) is set.

When the host pulls the $\overline{SS}$ line high, the transaction is done and the Transmit Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.TXC) will be set.

After DATA is written it takes up to three SCK clock cycles until the content of DATA is ready to be loaded into the Shift register on the next character boundary. As a consequence, the first character transferred in a SPI transaction will not be the content of DATA. This can be avoided by using the preloading feature. Refer to Preloading of the Client Shift Register.

When transmitting several characters in one SPI transaction, the data has to be written into DATA register with at least three SCK clock cycles left in the current character transmission. If this criteria is not met, the previously received character will be transmitted.

Once the DATA register is empty, it takes three CLK_SERCOM_APB cycles for INTFLAG.DRE to be set.

#### 35.6.2.7. Receiver Error Bit

The SPI receiver has one error bit: the Buffer Overflow bit (BUFOVF), which can be read from the Status register (STATUS). Once an error happens, the bit will stay set until it is cleared by writing '1' to it. The bit is also automatically cleared when the receiver is disabled.

There are two methods for buffer overflow notification, selected by the immediate Buffer Overflow Notification bit in the Control A register (CTRLA.IBON):

If CTRLA.IBON=1, STATUS.BUFOVF is raised immediately upon buffer overflow. Software can then empty the receive FIFO by reading RxDATA until the receiver complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) goes low.

If CTRLA.IBON=0, the Buffer Overflow condition travels with data through the receive FIFO. After the received data is read, STATUS.BUFOVF and INTFLAG.ERROR will be set along with INTFLAG.RXC, and RxDATA will be zero.

### 35.6.3. Additional Features

#### 35.6.3.1. Address Recognition

When the SPI is configured for client operation (CTRLA.MODE=0x2) with address recognition (CTRLA.FORM is 0x2), the SERCOM address recognition logic is enabled: the first character in a transaction is checked for an address match.

If there is a match, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set, the MISO output is enabled, and the transaction is processed. If the device is in Sleep mode, an address match can wake-up the device in order to process the transaction.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

If there is no match, the complete transaction is ignored.

If a 9-bit frame format is selected, only the lower 8 bits of the Shift register are checked against the Address register (ADDR).

Preload must be disabled (CTRLB.PLOADEN=0) in order to use this mode.

**Related Links**

Address Match and Mask

### 35.6.3.2. Preloading of the Client Shift Register

When starting a transaction, the client will first transmit the contents of the shift register before loading new data from DATA. The first character sent can be either the reset value of the shift register (if this is the first transmission since the last reset) or the last character in the previous transmission.

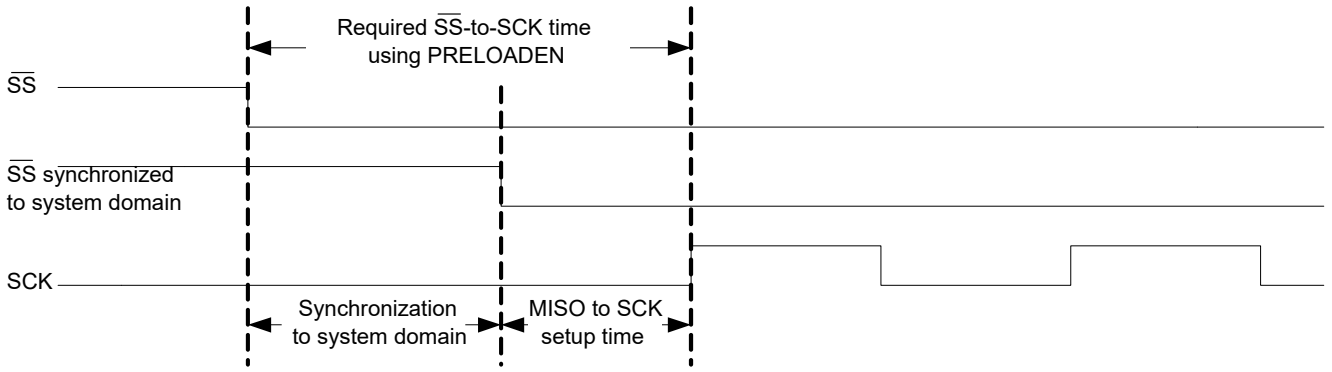
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

Only one data character will be preloaded into the shift register while the synchronized $\overline{SS}$ signal is high. If the next character is written to DATA before $\overline{SS}$ is pulled low, the second character will be stored in DATA until transfer begins.

For proper preloading, sufficient time must elapse between $\overline{SS}$ going low and the first SCK sampling edge, as in Timing Using Preloading. See also the *Electrical Characteristics* chapters for timing details.

Preloading is enabled by writing '1' to the Client Data Preload Enable bit in the CTRLB register (CTRLB.PLOADEN).

**Figure 35-4.** Timing Using Preloading

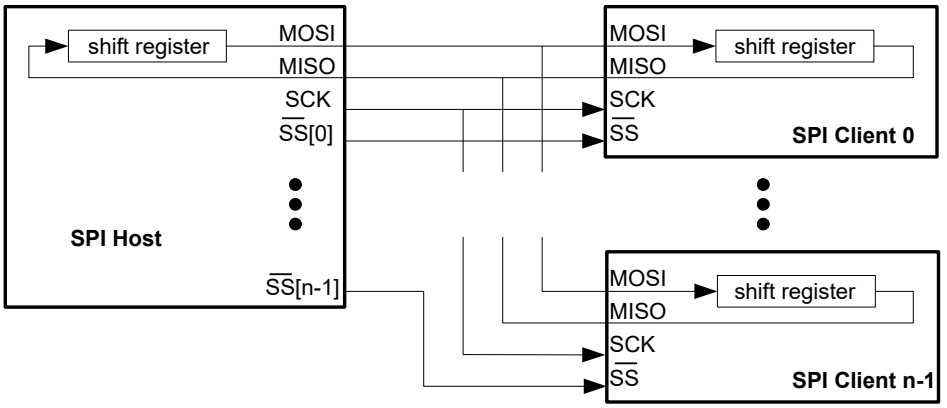



### 35.6.3.3. Host with Several Clients

Host with multiple clients in parallel is only available when Host SPI Select Enable (CTRLB.MSSEN) is set to zero and hardware $\overline{SS}$ control is disabled. If the bus consists of several SPI clients, a SPI host can use general purpose I/O pins to control the $\overline{SS}$ line to each of the clients on the bus, as shown in the following figure. In this configuration, the single selected SPI client will drive the tri-state MISO line.

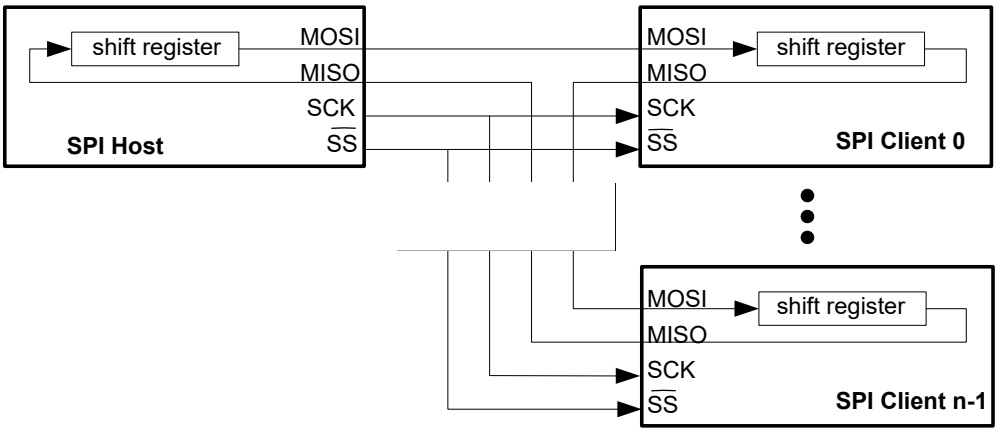

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 35-5.** Multiple Clients in Parallel




Another configuration is multiple clients in series, as shown in the following figure. In this configuration, all n attached clients are connected in series. A common $\overline{SS}$ line is provided to all clients, enabling them simultaneously. The host must shift n characters for a complete transaction. Depending on the Host SPI Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

**Figure 35-6.** Multiple Clients in Series




### 35.6.3.4. Loop-Back Mode

For Loop-back mode, configure the Data In Pinout (CTRLA.DIPO) and Data Out Pinout (CTRLA.DOPO) to use the same data pins for transmit and receive. The loop-back is through the pad, so the signal is also available externally.

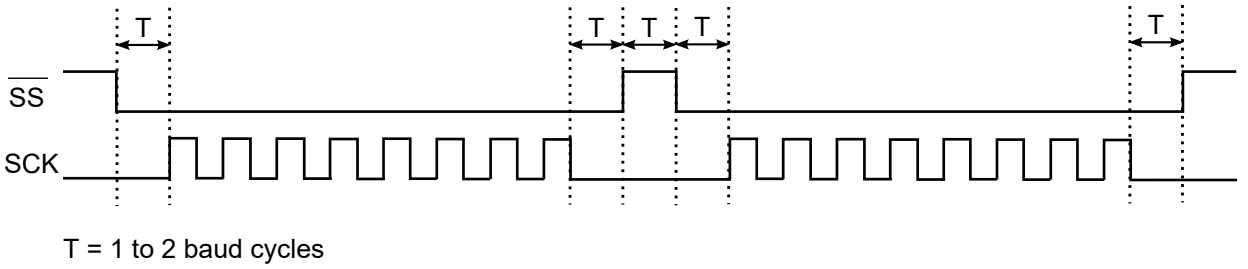
### 35.6.3.5. Hardware Controlled $\overline{SS}$

In Host mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Host SPI Select Enable (CTRLB.MSSEN) bit to '1'. In this mode, the $\overline{SS}$ pin is driven low for a minimum of one baud cycle before transmission begins, and stays low for a minimum of one baud cycle after transmission completes. If back-to-back frames are transmitted, the $\overline{SS}$ pin will always be driven high for a minimum of one baud cycle between frames.

In Hardware Controlled $\overline{SS}$, the time T is between one and two baud cycles depending on the SPI Transfer mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 35-7.** Hardware Controlled $\overline{SS}$




When CTRLB.MSSEN=0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

### 35.6.3.6. SPI Select Low Detection

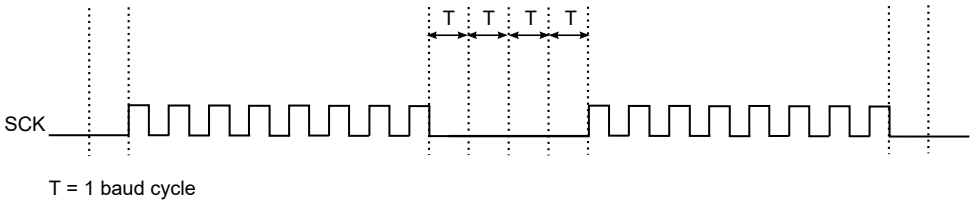
In Client mode, the SPI can wake the CPU when the SPI Select ($\overline{SS}$) goes low. When the SPI Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the SPI Select Low Interrupt flag (INTFLAG.SSL) and the device will wake-up if applicable.

### 35.6.3.7. Host Inter-Character Spacing

When configured as Host, inter-character spacing can be increased by writing a non-zero value to the Inter-Character Spacing bit field in the Control C register (CTRLC.ICSPACE). When non-zero, CTRLC.ICSPACE represents the minimum number of baud cycles that the SCK clock line does not toggle and the next character is stalled.

The figure gives an example for CTRLC.ICSPACE=4; In this case, the SCK is inactive for 4 baud cycles.

**Figure 35-8.** Four Cycle Inter-Character Spacing Example




### 35.6.3.8. 32-bit Extension

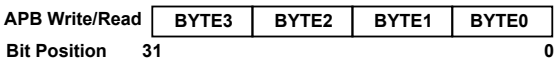
For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B=1). When enabled, write and read transaction to/from the DATA register are 32 bit in size.

If frames are not multiples of 4 Bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The figure below shows the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0 to 3.

Only 8-bit character size is supported.

**Figure 35-9.** 32-bit Extension Byte Ordering



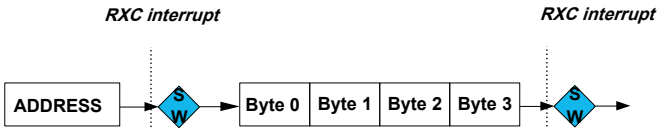

#### 32-bit Extension Client Operation

The figure below shows a transaction with 32-bit Extension enabled (CTRLC.DATA32B=1). When address recognition is enabled (CTRLA.FORM=0x2) and there is an address match, the address is loaded into the FIFO as Byte zero and data begins with Byte 1. INTFLAGS.RXC will then be raised for


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

every 4 Bytes transferred. For transmit, there is a 32-bit holding buffer in the core domain. Once DATA has been registered in the core domain, INTFLAG.DRE will be raised, so that the next 32 bits can be written to the DATA register.

**Figure 35-10.** 32-bit Extension Client Operation




When utilizing the length counter, the LENGTH register must be written before the frame begins. If the frame length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN Bytes, the Length Error Status bit (STATUS.LENERR) is raised. If LENGTH.LEN is not a multiple of 4 Bytes, the final INTFLAG.RXC interrupt will be raised when the last Byte is received.

The length count is based on the received Bytes, or the number of clocks if the receiver is not enabled. If pre-loading is disabled and DATA is written to for transmit before SCK starts, transmitted data will be delayed by one Byte, but the length counter will still increment for the first (empty) Byte transmission. When the counter reaches LENGTH.LEN, the internal length counter, Rx Byte counter, and Tx Byte counter are reset. If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.

If there is a Length Error (STATUS.LENERR), the remaining Bytes in the length will be transmitted at the beginning of the next frame. If this is not desired, the SERCOM must be disabled and re-enabled in order to flush the Tx and Rx pipelines.

Writing the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not configured and a frame is not a multiple of 4 Bytes (while $\overline{SS}$ is low), the remainder will be transmitted in the next frame.

### 32-bit Extension Host Operation

When using the SPI configured as Host, the Length and the Length Enable bit fields (LENGTH.LEN and LENGTH.LENEN) must be written before the frame begins. When LENGTH.LENEN is written to '1', the value of LENGTH.LEN determines the number of data bytes in the transaction from 1 to 255.

For receive data, INTFLAG.RXC is raised every 4 Bytes received. If LENGTH.LEN is not a multiple of 4 Bytes, the final INTFLAG.RXC is set when the final byte is received.

For transmit, there is a holding buffer for the 32-bit data in the core domain. Once DATA has been registered in the core domain, INTFLAG.DRE will be raised so that the next 32 bits can be written to the DATA register.

If multiple lengths are to be transmitted, INTFLAG.TXC must go high before writing DATA for subsequent lengths.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.6.4. DMA, Interrupts, and Events

**Table 35-4.** Module Request for SERCOM SPI

| Condition | Request | | |
|---|---|---|---|
| | DMA | Interrupt | Event |
| Data Register Empty (DRE) | Yes<br>(request cleared when data is written) | Yes | NA |
| Receive Complete (RXC) | Yes<br>(request cleared when data is read) | Yes | |
| Transmit Complete (TXC) | NA | Yes | |
| SPI Select low (SSL) | NA | Yes | |
| Error (ERROR) | NA | Yes | |

#### 35.6.4.1. DMA Operation

The SPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty. The request is cleared when DATA is written.

#### 35.6.4.2. Interrupts

The SPI has the following interrupt sources. These are asynchronous interrupts, and can wake-up the device from any Sleep mode:

- Data Register Empty (DRE)
- Receive Complete (RXC)
- Transmit Complete (TXC)
- SPI Select Low (SSL)
- Error (ERROR)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is met. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the Interrupt flag is set and if the corresponding interrupt is enabled. The interrupt request remains active until either the Interrupt flag is cleared, the interrupt is disabled, or the SPI is reset. For details on clearing Interrupt flags, refer to the INTFLAG register description.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller

#### 35.6.4.3. Events

Not applicable.

### 35.6.5. Sleep Mode Operation

The behavior in Sleep mode is depending on the host/client configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Host operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_SERCOMx_CORE will continue to run in Idle Sleep mode and in Standby Sleep mode. Any interrupt can wake-up the device.
- Host operation, CTRLA.RUNSTDBY=0: GLK_SERCOMx_CORE will be disabled after the ongoing transaction is finished. Any interrupt can wake up the device.
- Client operation, CTRLA.RUNSTDBY=1: The Receive Complete interrupt can wake-up the device.
- Client operation, CTRLA.RUNSTDBY=0: All reception will be dropped, including the ongoing transaction.

### 35.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

**Note:** CTRLB.RXEN is write-synchronized somewhat differently. See also *CTRLB* register for details.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 35.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | IBON |
| | | 23:16 | | | DIPO[1:0] | | | | DOPO[1:0] | |
| | | 31:24 | | DORD | CPOL | CPHA | FORM[3:0] | | | |
| 0x04 | CTRLB | 7:0 | | PLOADEN | | | | CHSIZE[2:0] | | |
| | | 15:8 | AMODE[1:0] | | MSSEN | | | | SSDE | |
| | | 23:16 | | | | | | | RXEN | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | ICSPACE[5:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | DATA32B |
| 0x0C | BAUD | 7:0 | BAUD[7:0] | | | | | | | |
| 0x0D ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | ERROR | | | | SSL | RXC | TXC | DRE |
| 0x15 | Reserved | | | | | | | | | |
| 0x16 | INTENSET | 7:0 | ERROR | | | | SSL | RXC | TXC | DRE |
| 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTFLAG | 7:0 | ERROR | | | | SSL | RXC | TXC | DRE |
| 0x19 | Reserved | | | | | | | | | |
| 0x1A | STATUS | 7:0 | | | | | | BUFOVF | | |
| | | 15:8 | | | | | LENERR | | | |
| 0x1C | SYNCBUSY | 7:0 | | | | LENGTH | | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 ... 0x21 | Reserved | | | | | | | | | |
| 0x22 | LENGTH | 7:0 | LEN[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | LENEN |
| 0x24 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | ADDRMASK[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | DATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x2C ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DBGCTRL | 7:0 | | | | | | | | DBGSTOP |

## 35.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Refer to Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Refer to Register Access Protection.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | DORD | CPOL | CPHA | FORM[3:0] | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | DIPO[1:0] | | | | DOPO[1:0] | |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | IBON |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 30 – DORD** Data Order
This bit selects the data order when a character is shifted out from the Shift register.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | MSB is transferred first. |
| 1 | LSB is transferred first. |

**Bit 29 – CPOL** Clock Polarity
In combination with the Clock Phase bit (CPHA), this bit determines the SPI Transfer mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | SCK is low when idle. The leading edge of a clock cycle is a rising edge, while the trailing edge is a falling edge. |
| 1 | SCK is high when idle. The leading edge of a clock cycle is a falling edge, while the trailing edge is a rising edge. |

**Bit 28 – CPHA** Clock Phase
In combination with the Clock Polarity bit (CPOL), this bit determines the SPI Transfer mode.
This bit is not synchronized.

| Mode | CPOL | CPHA | Leading Edge | Trailing Edge |
|---|---|---|---|---|
| 0x0 | 0 | 0 | Rising, sample | Falling, change |
| 0x1 | 0 | 1 | Rising, change | Falling, sample |
| 0x2 | 1 | 0 | Falling, sample | Rising, change |
| 0x3 | 1 | 1 | Falling, change | Rising, sample |

| Value | Description |
|---|---|
| 0 | The data is sampled on a leading SCK edge and changed on a trailing SCK edge. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The data is sampled on a trailing SCK edge and changed on a leading SCK edge. |

**Bits 27:24 – FORM[3:0]** Frame Format

This bit field selects the various frame formats supported by the SPI in Client mode. When the 'SPI frame with address' format is selected, the first byte received is checked against the ADDR register.

| FORM[3:0] | Name | Description |
|---|---|---|
| 0x0 | SPI | SPI frame |
| 0x1 | - | Reserved |
| 0x2 | SPI_ADDR | SPI frame with address |
| 0x3-0xF | - | Reserved |

**Bits 21:20 – DIPO[1:0]** Data In Pinout

These bits define the Data In (DI) pad configurations.
In Host operation, DI is MISO.
In Client operation, DI is MOSI.
These bits are not synchronized.

| DIPO[1:0] | Name | Description |
|---|---|---|
| 0x0 | PAD[0] | SERCOM PAD[0] is used as data input |
| 0x1 | PAD[1] | SERCOM PAD[1] is used as data input |
| 0x2 | PAD[2] | SERCOM PAD[2] is used as data input |
| 0x3 | PAD[3] | SERCOM PAD[3] is used as data input |

**Bits 17:16 – DOPO[1:0]** Data Out Pinout

This bit defines the available pad configurations for Data Out (DO) and the Serial Clock (SCK). In Client operation, the SPI Select ($\overline{SS}$) line is controlled by DOPO, while in Host operation the $\overline{SS}$ line is controlled by the port configuration.
In Host operation, DO is MOSI.
In Client operation, DO is MISO.
These bits are not synchronized.

| DOPO | DO | SCK | Client $\overline{SS}$ | Host $\overline{SS}$ |
|---|---|---|---|---|
| 0x0 | PAD[0] | PAD[1] | PAD[2] | PAD[2] Host $\overline{SS}$ pin when MSSEN = 1 otherwise System configuration |
| 0x1 | Reserved | | | |
| 0x2 | PAD[3] | PAD[1] | PAD[2] | PAD[2] Host $\overline{SS}$ pin when MSSEN = 1 otherwise System configuration |
| 0x3 | Reserved | | | |

**Bit 8 – IBON** Immediate Buffer Overflow Notification

This bit controls when the Buffer Overflow Status bit (STATUS.BUFOVF) is set when a buffer overflow occurs.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | STATUS.BUFOVF is set when it occurs in the data stream. |
| 1 | STATUS.BUFOVF is set immediately upon buffer overflow. |

**Bit 7 – RUNSTDBY** Run In Standby

This bit defines the functionality in Standby Sleep mode.
These bits are not synchronized.

| RUNSTDBY | Client | Host |
|---|---|---|
| 0x0 | Disabled. All reception is dropped, including the ongoing transaction. | Generic clock is disabled when ongoing transaction is finished. All interrupts can wake-up the device. |
| 0x1 | Ongoing transaction continues, wake on Receive Complete interrupt. | Generic clock is enabled while in sleep modes. All interrupts can wake-up the device. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 4:2 – MODE[2:0]** Operating Mode
These bits must be written to 0x2 or 0x3 to select the SPI of the SERCOM.
0x2: SPI Client operation
0x3: SPI Host operation
These bits are not synchronized.

**Bit 1 – ENABLE** Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the operation is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled or being disabled. |
| 1 | The peripheral is enabled or being enabled. |

**Bit 0 – SWRST** Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing ''1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing Reset will result in an APB error. Reading any register will return the Reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY. SWRST will both be cleared when the Reset is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | There is no Reset operation ongoing. |
| 1 | The Reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RXEN | |
| Access | | | | | | | R/W | |
| Reset | | | | | | | 0 | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | AMODE[1:0] | | MSSEN | | | | SSDE | |
| Access | R/W | R/W | R/W | | | | R/W | |
| Reset | 0 | 0 | 0 | | | | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | PLOADEN | | | | CHSIZE[2:0] | | |
| Access | | R/W | | | | R/W | R/W | R/W |
| Reset | | 0 | | | | 0 | 0 | 0 |

**Bit 17 – RXEN** Receiver Enable

Writing '0' to this bit will disable the SPI receiver immediately. The receive buffer will be flushed, data from ongoing receptions will be lost and STATUS.BUFOVF will be cleared.
Writing '1' to CTRLB.RXEN when the SPI is disabled will set CTRLB.RXEN immediately. When the SPI is enabled, CTRLB.RXEN will be cleared, SYNCBUSY.CTRLB will be set and remain set until the receiver is enabled. When the receiver is enabled CTRLB.RXEN will read back as '1'.
Writing '1' to CTRLB.RXEN when the SPI is enabled will set SYNCBUSY.CTRLB, which will remain set until the receiver is enabled, and CTRLB.RXEN will read back as '1'.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The receiver is disabled or being enabled. |
| 1 | The receiver is enabled or it will be enabled when SPI is enabled. |

**Bits 15:14 – AMODE[1:0]** Address Mode

These bits set the Client Addressing mode when the frame format (CTRLA.FORM) with address is used. They are unused in Host mode.
These bits are not synchronized.

| AMODE[1:0] | Name | Description |
|---|---|---|
| 0x0 | MASK | ADDRMASK is used as a mask to the ADDR register |
| 0x1 | 2_ADDRS | The client responds to the two unique addresses in ADDR and ADDRMASK |
| 0x2 | RANGE | The client responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit |
| 0x3 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 13 – MSSEN** Host SPI Select Enable

This bit enables hardware SPI Select ($\overline{SS}$) control.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Hardware $\overline{SS}$ control is disabled. |
| 1 | Hardware $\overline{SS}$ control is enabled. |

**Bit 9 – SSDE** SPI Select Low Detect Enable

This bit enables wake-up when the SPI Select ($\overline{SS}$) pin transitions from high to low.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | $\overline{SS}$ low detector is disabled. |
| 1 | $\overline{SS}$ low detector is enabled. |

**Bit 6 – PLOADEN** Client Data Preload Enable

Setting this bit will enable preloading of the Client Shift register when there is no transfer in progress. If the $\overline{SS}$ line is high when DATA is written, it will be transferred immediately to the Shift register.
This bit is not synchronized.

**Bits 2:0 – CHSIZE[2:0]** Character Size

These bits are not synchronized.

| CHSIZE[2:0] | Name | Description |
|---|---|---|
| 0x0 | 8BIT | 8 bits |
| 0x1 | 9BIT | 9 bits |
| 0x2-0x7 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DATA32B |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | ICSPACE[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 24 – DATA32B** Data 32 Bit

This bit enables 32-bit Extension for read and write transactions to the DATA register.
When disabled, access is according to CTRLB.CHSIZE.

| Value | Description |
|---|---|
| 0 | Transactions from and to DATA register are 8-bit |
| 1 | Transactions from and to DATA register are 32-bit |

**Bits 5:0 – ICSPACE[5:0]** Inter-Character Spacing

When non-zero, CTRLC.ICSPACE selects the minimum number of baud cycles the SCK line will not toggle between characters.

| Value | Description |
|---|---|
| 0x00 | Inter-Character Spacing is disabled |
| 0x01-0x3F | The minimum Inter-Character Spacing |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.4. Baud Rate

**Name:** BAUD
**Offset:** 0x0C
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – BAUD[7:0]** Baud Register
These bits control the clock generation, as described in the *SERCOM Clock Generation – Baud-Rate Generator*.

**Related Links**
Clock Generation – Baud-Rate Generator


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | SSL | RXC | TXC | DRE |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 3 – SSL** SPI Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the SPI Select Low Interrupt Enable bit, which disables the SPI Select Low interrupt.

| Value | Description |
|---|---|
| 0 | SPI Select Low interrupt is disabled. |
| 1 | SPI Select Low interrupt is enabled. |

**Bit 2 – RXC** Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Receive Complete Interrupt Enable bit, which disables the Receive Complete interrupt.

| Value | Description |
|---|---|
| 0 | Receive Complete interrupt is disabled. |
| 1 | Receive Complete interrupt is enabled. |

**Bit 1 – TXC** Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Transmit Complete Interrupt Enable bit, which disable the Transmit Complete interrupt.

| Value | Description |
|---|---|
| 0 | Transmit Complete interrupt is disabled. |
| 1 | Transmit Complete interrupt is enabled. |

**Bit 0 – DRE** Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.

| Value | Description |
|---|---|
| 0 | Data Register Empty interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Data Register Empty interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x16
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | SSL | RXC | TXC | DRE |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 3 – SSL** SPI Select Low Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the SPI Select Low Interrupt Enable bit, which enables the SPI Select Low interrupt.

| Value | Description |
|---|---|
| 0 | SPI Select Low interrupt is disabled. |
| 1 | SPI Select Low interrupt is enabled. |

**Bit 2 – RXC** Receive Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Receive Complete Interrupt Enable bit, which enables the Receive Complete interrupt.

| Value | Description |
|---|---|
| 0 | Receive Complete interrupt is disabled. |
| 1 | Receive Complete interrupt is enabled. |

**Bit 1 – TXC** Transmit Complete Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Transmit Complete Interrupt Enable bit, which enables the Transmit Complete interrupt.

| Value | Description |
|---|---|
| 0 | Transmit Complete interrupt is disabled. |
| 1 | Transmit Complete interrupt is enabled. |

**Bit 0 – DRE** Data Register Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Data Register Empty Interrupt Enable bit, which enables the Data Register Empty interrupt.

| Value | Description |
|---|---|
| 0 | Data Register Empty interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Data Register Empty interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | SSL | RXC | TXC | DRE |
| Access | R/W | | | | R/W | R | R/W | R |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Error
This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that will set this flag have corresponding Status flags in the STATUS register. The BUFOVF error and the LENERR error will set this Interrupt flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 3 – SSL** SPI Select Low
This flag is cleared by writing '1' to it.
This bit is set when a high to low transition is detected on the $\overline{SS}$ pin in Client mode and SPI Select Low Detect (CTRLB.SSDE) is enabled.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 2 – RXC** Receive Complete
This flag is cleared by reading the Data (DATA) register or by disabling the receiver.
This flag is set when there are unread data in the receive buffer. If address matching is enabled, the first data received in a transaction will be an address.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.

**Bit 1 – TXC** Transmit Complete
This flag is cleared by writing '1' to it or by writing new data to DATA.
In Host mode, this flag is set when the data have been shifted out and there are no new data in DATA.
In Client mode, this flag is set when the $\overline{SS}$ pin is pulled high. If address matching is enabled, this flag is only set if the transaction was initiated with an address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 0 – DRE** Data Register Empty
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready for new data to transmit.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.8. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** –

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LENERR | | | |
| Access | | | | | R/W | | | |
| Reset | | | | | 0 | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | BUFOVF | | |
| Access | | | | | | R/W | | |
| Reset | | | | | | 0 | | |

**Bit 11 – LENERR** Transaction Length Error
This bit is set in Client mode when the length counter is enabled (LENGTH.LENEN=1) and the transfer length while $\overline{SS}$ is low is not a multiple of LENGTH.LEN.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

| Value | Description |
|---|---|
| 0 | No Length Error has occurred. |
| 1 | A Length Error has occurred. |

**Bit 2 – BUFOVF** Buffer Overflow
Reading this bit before reading DATA will indicate the error status of the next character to be read.
This bit is cleared by writing '1' to the bit or by disabling the receiver.
This bit is set when a Buffer Overflow condition is detected. See also CTRLA.IBON for overflow handling.
When set, the corresponding RxDATA will be zero.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.

| Value | Description |
|---|---|
| 0 | No Buffer Overflow has occurred. |
| 1 | A Buffer Overflow has occurred. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.9. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | LENGTH | | CTRLB | ENABLE | SWRST |
| Access | | | | R | | R | R | R |
| Reset | | | | 0 | | 0 | 0 | 0 |

**Bit 4 – LENGTH** LENGTH Synchronization Busy

Writing to the LENGTH register requires synchronization. When writing to LENGTH, SYNCBUSY.LENGTH will be set until synchronization is complete. If the LENGTH register is written to while SYNCBUSY.LENGTH is asserted, an APB error is generated.

**Note:** In Client mode, the clock is only running during data transfer, so SYNCBUSY.LENGTH will remain asserted until the next data transfer begins.

| Value | Description |
|---|---|
| 0 | LENGTH synchronization is not busy. |
| 1 | LENGTH synchronization is busy. |

**Bit 2 – CTRLB** CTRLB Synchronization Busy

Writing to the CTRLB when the SERCOM is enabled requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.CTRLB=1 until synchronization is complete. If CTRLB is written while SYNCBUSY.CTRLB=1, an APB error will be generated.

| Value | Description |
|---|---|
| 0 | CTRLB synchronization is not busy. |
| 1 | CTRLB synchronization is busy. |

**Bit 1 – ENABLE** SERCOM Enable Synchronization Busy

Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.ENABLE=1 until synchronization is complete.

| Value | Description |
|---|---|
| 0 | Enable synchronization is not busy. |
| 1 | Enable synchronization is busy. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – SWRST** Software Reset Synchronization Busy

Resetting the SERCOM (CTRLA.SWRST) requires synchronization. Ongoing synchronization is indicated by SYNCBUSY.SWRST=1 until synchronization is complete.

| Value | Description |
|---|---|
| 0 | SWRST synchronization is not busy. |
| 1 | SWRST synchronization is busy. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.10. Length

**Name:** LENGTH
**Offset:** 0x22
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | LENEN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LEN[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 8 – LENEN** Data Length Enable
In 32-bit Extension mode, this bit field enables the length counter.

| Value | Description |
|---|---|
| 0 | Length counter disabled |
| 1 | Length counter enabled |

**Bits 7:0 – LEN[7:0]** Data Length
In 32-bit Extension mode, this bit field configures the data length after which the flags INTFLAG.RCX or INTFLAG.DRE are raised.

| Value | Description |
|---|---|
| 0x00 | Reserved if LENEN=0x1 |
| 0x01-0xFF | Data Length |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.11. Address

**Name:** ADDR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDRMASK[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – ADDRMASK[7:0]** Address Mask
These bits hold the address mask when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).

**Bits 7:0 – ADDR[7:0]** Address
These bits hold the address when the transaction format with address is used (CTRLA.FORM, CTRLB.AMODE).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.12. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x0000
**Property:** –

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Data

Reading these bits will return the contents of the receive data buffer. The register should be read only when the Receive Complete Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.RXC) is set.

Writing these bits will write the transmit data buffer. This register should be written only when the Data Register Empty Interrupt Flag bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is set.

Reads and writes are 32-bit or CTLB.CHSIZE based on the CTRLC.DATA32B setting.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.8.13. Debug Control

**Name:** DBGCTRL
**Offset:** 0x30
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGSTOP |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGSTOP** Debug Stop Mode
This bit controls the functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The baud-rate generator continues normal operation when the CPU is halted by an external debugger. |
| 1 | The baud-rate generator is halted when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 36. SERCOM I²C – Inter-Integrated Circuit

## 36.1. Overview

The Inter-Integrated Circuit (I²C) interface is one of the available modes in the Serial Communication Interface (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in Figure 36-1. Labels in capital letters are registers accessible by the CPU, while lowercase labels are internal to the SERCOM.

A SERCOM instance can be configured to be either an I²C Host or an I²C Client. Both Host and Client have an interface containing a Shift register, a transmit buffer and a receive buffer. In addition, the I²C Host uses the SERCOM baud-rate generator, while the I²C Client uses the SERCOM address match logic.

**Related Links**

SERCOM – Serial Communication Interface

## 36.2. Features

SERCOM I²C includes the following features:

- Host or Client Operation
- Can be used with DMA
- Philips I²C Compatible
- SMBus Compatible
- PMBus™ Compatible
- Support of 100 kHz and 400 kHz, 1 MHz and 3.4 MHz I²C mode
- 32-bit Data Extension for better system bus utilization
- 4-Wire Operation Supported
- Physical interface includes:
  - Slew-rate limited outputs
  - Filtered inputs
- Client Operation:
  - Operation in all Sleep modes
  - Wake-up on address match
  - 7-bit and 10-bit Address match in hardware for:
  - - Unique address and/or 7-bit general call address
    - Address range
    - Two unique addresses can be used with DMA

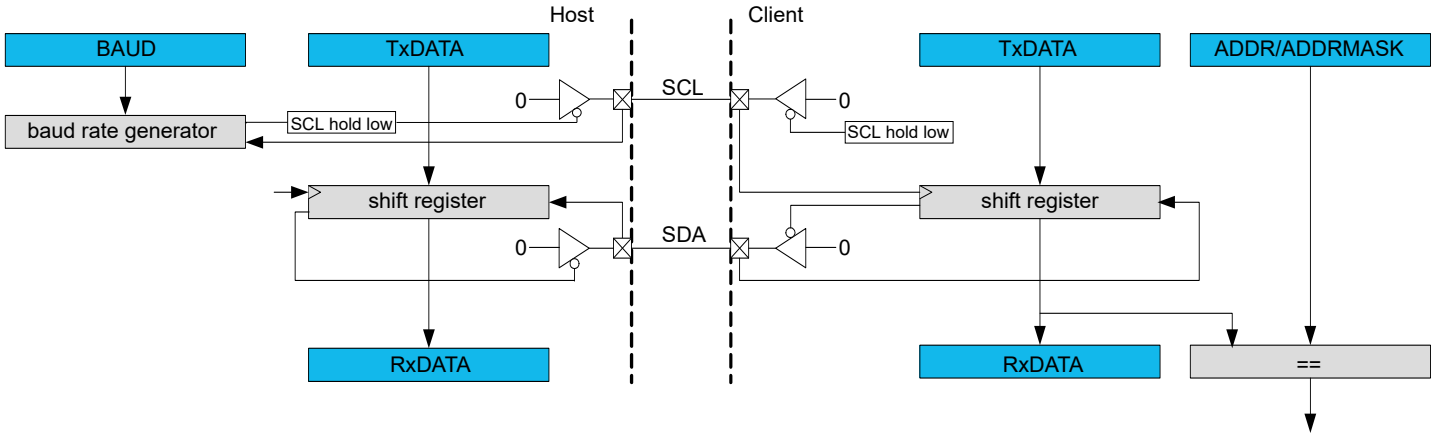
**Related Links**

Features


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 36.3. Block Diagram

**Figure 36-1.** I²C Single-Host Single-Client Interconnection




## 36.4. Signal Description

| Signal Name | Type | Description |
|---|---|---|
| PAD[0] | Digital I/O | SDA |
| PAD[1] | Digital I/O | SCL |
| PAD[2] | Digital I/O | SDA_OUT (4-wire operation) |
| PAD[3] | Digital I/O | SCL_OUT (4-wire operation) |

One signal can be mapped on several pins.

Not all the pins are I²C pins. Refer to SERCOM I2C Configurations table for additional information.

**Related Links**
I/O Multiplexing and Considerations
SERCOM I2C Configurations
4-Wire Mode

## 36.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 36.5.1. I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

When the SERCOM is used in I²C mode, the SERCOM controls the direction and value of the I/O pins. In I²C mode pull-up resistors are disabled. External pull-up resistors are required for proper function.

**Related Links**
PORT - I/O Pin Controller

### 36.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake-up the device from Sleep modes.

**Related Links**
PM – Power Manager


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.5.3. Clocks

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Main Clock Controller. Refer to *Peripheral Clock Masking* for details and default status of this clock.

Two generic clocks are used by SERCOM: GCLK_SERCOMx_CORE and GCLK_SERCOM_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the $I^2C$ when working as a host. The slow clock (GCLK_SERCOM_SLOW) is required only for certain functions, e.g. SMBus timing. These two clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the $I^2C$.

These generic clocks are asynchronous to the bus clock (CLK_SERCOMx_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**
GCLK - Generic Clock Controller
Peripheral Clock Masking
PM – Power Manager

### 36.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

**Related Links**
DMAC – Direct Memory Access Controller

### 36.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller

### 36.5.6. Events

Not applicable.

### 36.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar, improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

### 36.5.8. Register Access Protection

Registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC).

PAC write protection is not available for the following registers:

- Interrupt Flag Clear and Status register (INTFLAG)
- Status register (STATUS)
- Data register (DATA)
- Address register (ADDR)

Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Write-protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 36.5.9. Analog Connections

Not applicable.

## 36.6. Functional Description

### 36.6.1. Principle of Operation

The I²C interface uses two physical lines for communication:

- Serial Data Line (SDA) for data transfer
- Serial Clock Line (SCL) for the bus clock

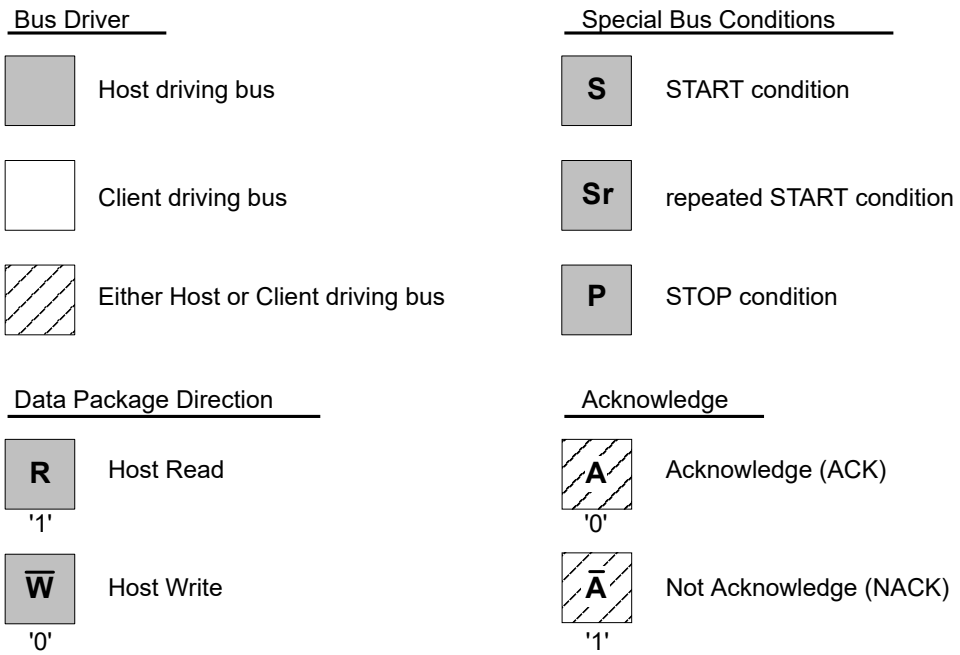
A transaction starts with the I²C host sending the Start condition, followed by a 7-bit address and a direction bit (read or write to/from the client).

The addressed I²C client will then Acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is Not Acknowledged (NACK), whether by the I²C client or host, the I²C host takes action by either terminating the transaction by sending the Stop condition, or by sending a repeated start to transfer more data.

The figure below illustrates the possible transaction formats and Transaction Diagram Symbols explains the transaction symbols. These symbols will be used in the following descriptions.

**Figure 36-2.** Transaction Diagram Symbols

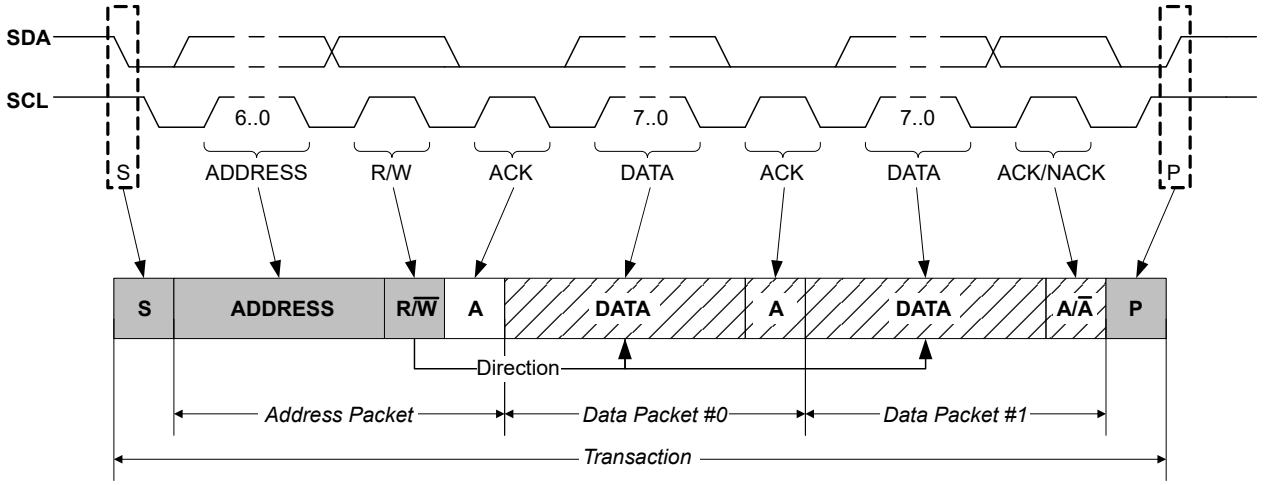




© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

**Figure 36-3.** Basic I²C Transaction Diagram




## 36.6.2. Basic Operation

### 36.6.2.1. Initialization

The following registers are enable-protected, meaning they can be written only when the I²C interface is disabled (CTRLA.ENABLE is '0'):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST) bits
- Control B register (CTRLB), except Acknowledge Action (CTRLB.ACKACT) and Command (CTRLB.CMD) bits
- Baud register (BAUD)
- Address register (ADDR) in client operation.

When the I²C is enabled or is being enabled (CTRLA.ENABLE=1), writing to these registers will be discarded. If the I²C is being disabled, writing to these registers will be completed after the disabling.

Enable-protection is denoted by the "Enable-Protection" property in the register description.

Before the I²C is enabled it must be configured as outlined by the following steps:

1. Select I²C Host or Client mode by writing 0x4 (Client mode) or 0x5 (Host mode) to the Operating Mode bits in the CTRLA register (CTRLA.MODE).
2. If desired, select the SDA Hold Time value in the CTRLA register (CTRLA.SDAHOLD).
3. In Client mode, the minimum client setup time for the SDA can be selected in the SDA Setup Time bit group in the Control C register (CTRLC.SDASETUP).
4. If desired, enable smart operation by setting the Smart Mode Enable bit in the CTRLB register (CTRLB.SMEN).
5. If desired, enable SCL low time-out by setting the SCL Low Time-Out bit in the Control A register (CTRLA.LOWTOUTEN).
6. In Host mode:
   a. Select the inactive bus time-out in the Inactive Time-Out bit group in the CTRLA register (CTRLA.INACTOUT).
   b. Write the Baud Rate register (BAUD) to generate the desired baud rate.

   In Client mode:
   a. Configure the address match configuration by writing the Address Mode value in the CTRLB register (CTRLB.AMODE).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

b. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

### 36.6.2.2. Enabling, Disabling, and Resetting

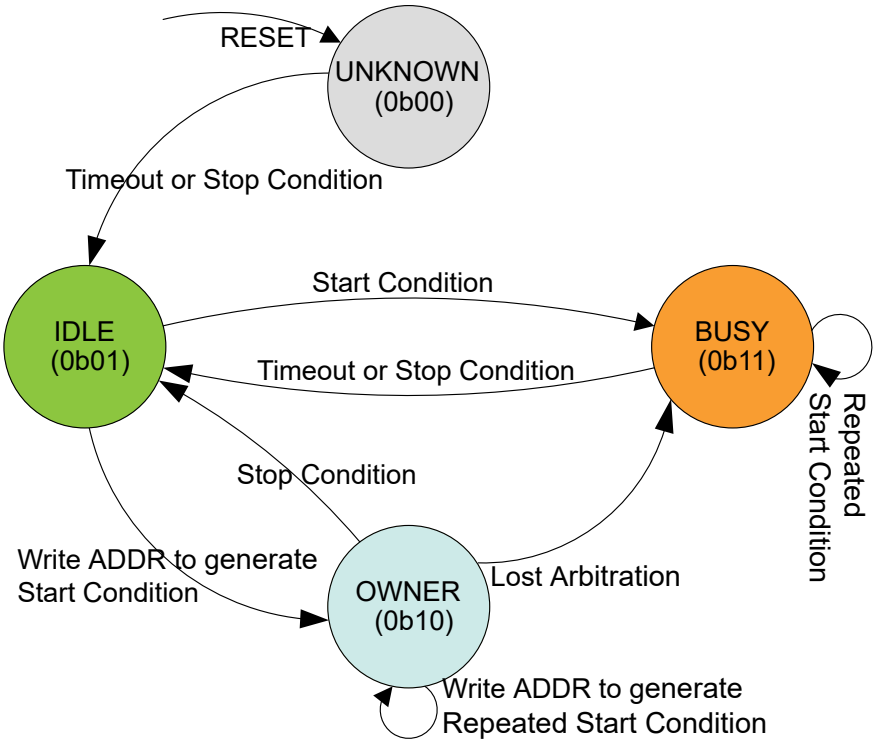
This peripheral is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE), and disabled by writing '0' to it.

Writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

### 36.6.2.3. I²C Bus State Logic

The Bus state logic includes several logic blocks that continuously monitor the activity on the I²C bus lines in all Sleep modes with running GCLK_SERCOM_x clocks. The start and stop detectors and the bit counter are all essential in the process of determining the current Bus state. The Bus state is determined according to Bus State Diagram. Software can get the current Bus state by reading the Host Bus State bits in the Status register (STATUS.BUSSTATE). The value of STATUS.BUSSTATE in the figure is shown in binary.

**Figure 36-4.** Bus State Diagram




The Bus state machine is active when the I²C host is enabled.

After the I²C host has been enabled, the Bus state is UNKNOWN (0b00). From the UNKNOWN state, the bus will transition to IDLE (0b01) by either:

- Forcing by writing 0b01 to STATUS.BUSSTATE
- A Stop condition is detected on the bus
- If the inactive bus time-out is configured for SMBus compatibility (CTRLA.INACTOUT) and a time-out occurs.

**Note:** Once a known Bus state is established, the Bus state logic will not re-enter the UNKNOWN state.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When the bus is IDLE it is ready for a new transaction. If a Start condition is issued on the bus by another I²C host in a multi-host setup, the bus becomes BUSY (0b11). The bus will re-enter IDLE either when a Stop condition is detected, or when a time-out occurs (inactive bus time-out needs to be configured).

If a Start condition is generated internally by writing the Address bit group in the Address register (ADDR.ADDR) while IDLE, the OWNER state (0b10) is entered. If the complete transaction was performed without interference, i.e., arbitration was not lost, the I²C host can issue a Stop condition, which will change the Bus state back to IDLE.

However, if a packet collision is detected while in OWNER state, the arbitration is assumed lost and the Bus state becomes BUSY until a Stop condition is detected. A repeated Start condition will change the Bus state only if arbitration is lost while issuing a repeated start.

**Note:** Violating the protocol may cause the I²C to hang. If this happens it is possible to recover from this state by a software Reset (CTRLA.SWRST='1').

**Related Links**

CTRLA

Control A

#### 36.6.2.4. I²C Host Operation

The I²C Host is byte-oriented and interrupt based. The number of interrupts generated is kept at a minimum by automatic handling of most incidents. The software driver complexity and code size are reduced by auto-triggering of operations, and a Special Smart mode, which can be enabled by the Smart Mode Enable bit in the Control A register (CTRLA.SMEN).

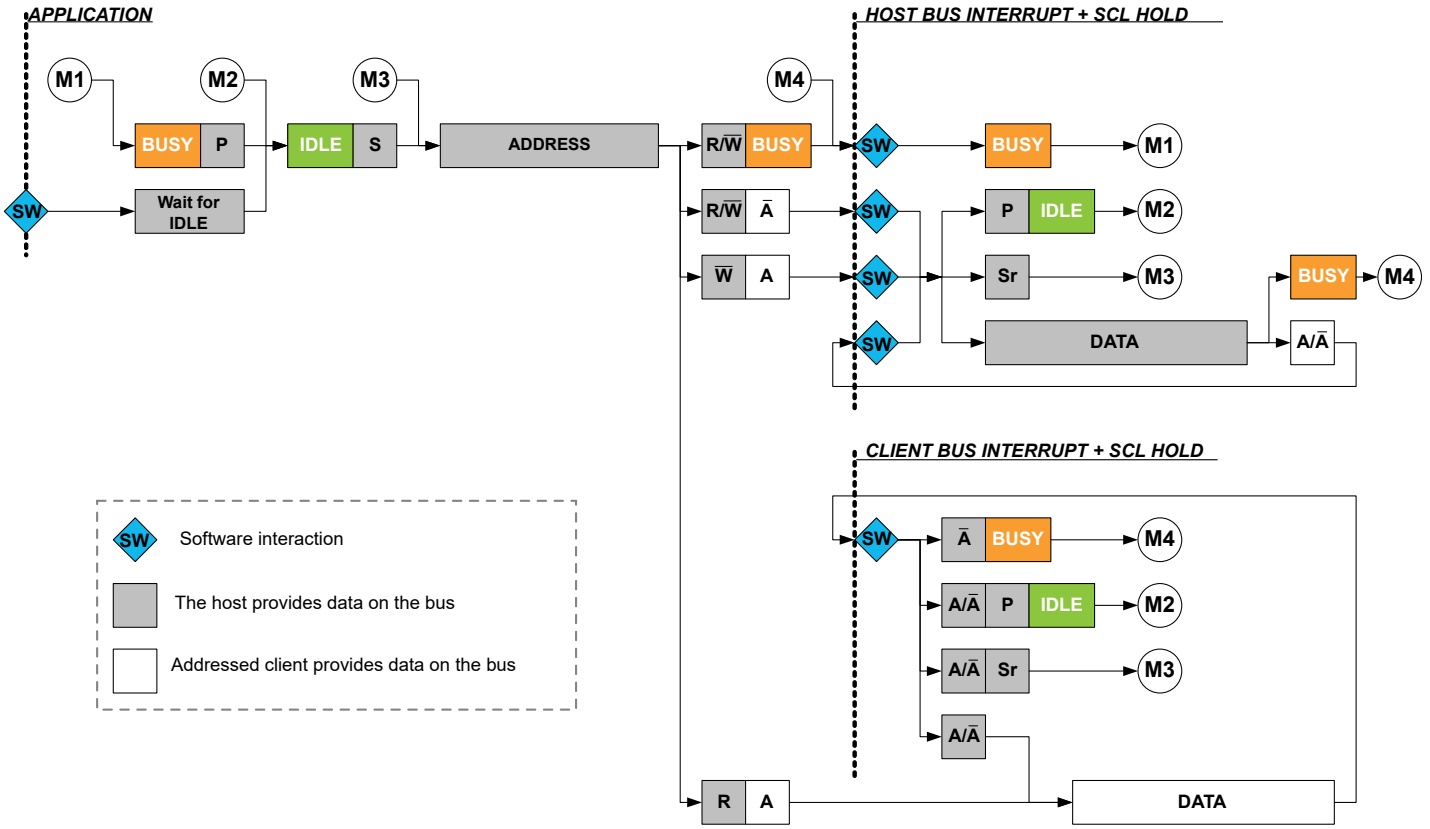
The I²C Host has two interrupt strategies.

When SCL Stretch Mode (CTRLA.SCLSM) is '0', SCL is stretched before or after the Acknowledge bit . In this mode the I²C Host operates according to Host Behavioral Diagram (SCLSM=0). The circles labeled "M*n*" (M1, M2..) indicate the nodes the bus logic can jump to, based on software or hardware interaction.

This diagram is used as reference for the description of the I²C Host operation throughout the document.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 36-5.** I²C Host Behavioral Diagram (SCLSM=0)

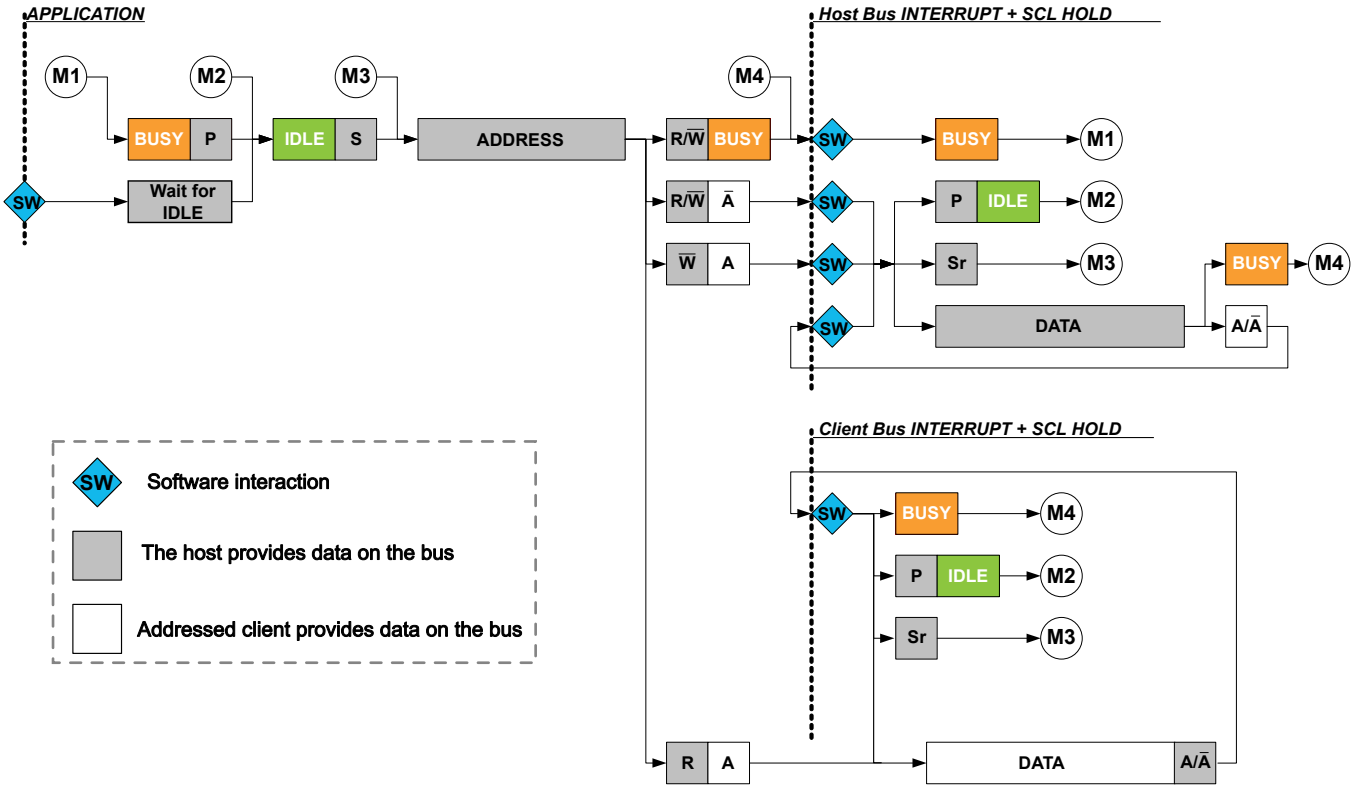



In the second strategy (CTRLA.SCLSM=1), interrupts only occur after the ACK bit, as in Host Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging.

**Note:** I²C High-speed (*Hs*) mode requires CTRLA.SCLSM=1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 36-6.** I²C Host Behavioral Diagram (SCLSM=1)




### 36.6.2.4.1. Host Clock Generation

The SERCOM peripheral supports several I²C bidirectional modes:

- Standard mode (*Sm*) up to 100 kHz
- Fast mode (*Fm*) up to 400 kHz
- Fast mode Plus (*Fm+*) up to 1 MHz
- High-speed mode (*Hs*) up to 3.4 MHz

The Host clock configuration for *Sm, Fm,* and *Fm+* are described in Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus). For *Hs*, refer to Host Clock Generation (High-Speed Mode).

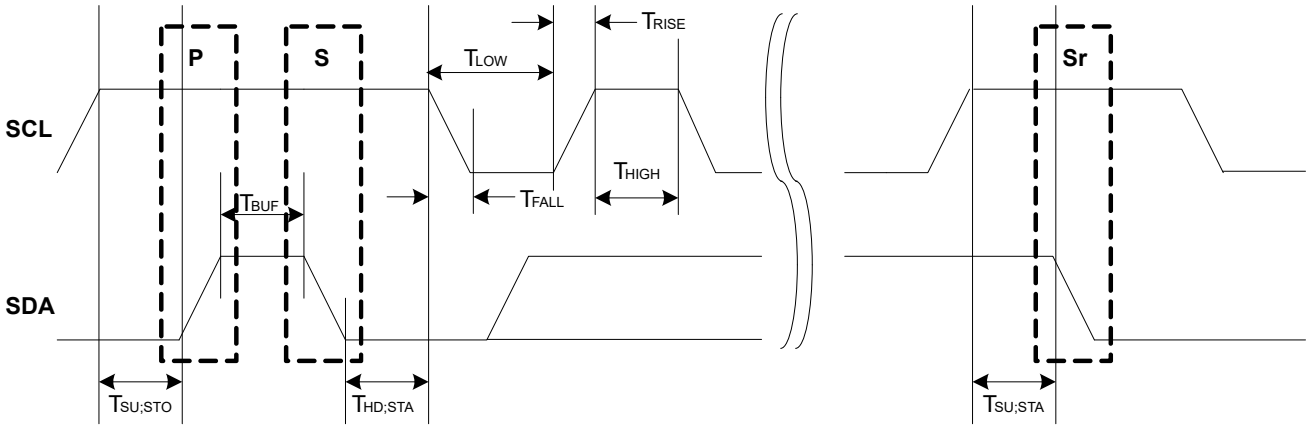
#### Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)

In I²C *Sm, Fm*, and *Fm+* mode, the Host clock (SCL) frequency is determined as described in this section:

The low ($T_{LOW}$) and high ($T_{HIGH}$) times are determined by the Baud Rate register (BAUD), while the rise ($T_{RISE}$) and fall ($T_{FALL}$) times are determined by the bus topology. Because of the wired-AND logic of the bus, $T_{FALL}$ will be considered as part of $T_{LOW}$. Likewise, $T_{RISE}$ will be in a state between $T_{LOW}$ and $T_{HIGH}$ until a high state has been detected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 36-7.** SCL Timing




The following parameters are timed using the SCL low time period $T_{LOW}$. This comes from the Host Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, or the Host Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- $T_{LOW}$ – Low period of SCL clock
- $T_{SU;STO}$ – Set-up time for stop condition
- $T_{BUF}$ – Bus free time between stop and start conditions
- $T_{HD;STA}$ – Hold time (repeated) start condition
- $T_{SU;STA}$ – Set-up time for repeated start condition
- $T_{HIGH}$ is timed using the SCL high time count from BAUD.BAUD
- $T_{RISE}$ is determined by the bus impedance; for internal pull-ups.
- $T_{FALL}$ is determined by the open-drain current limit and bus impedance; can typically be regarded as zero.

The SCL frequency is given by:

$$f_{SCL} = \frac{1}{T_{LOW} + T_{HIGH} + T_{RISE}}$$

When BAUD.BAUDLOW is zero, the BAUD.BAUD value is used to time both SCL high and SCL low. In this case the following formula will give the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + 2BAUD + f_{GCLK} \cdot T_{RISE}}$$

When BAUD.BAUDLOW is non-zero, the following formula determines the SCL frequency:

$$f_{SCL} = \frac{f_{GCLK}}{10 + BAUD + BAUDLOW + f_{GCLK} \cdot T_{RISE}}$$

The following formulas can determine the SCL $T_{LOW}$ and $T_{HIGH}$ times:

$$T_{LOW} = \frac{BAUDLOW + 5}{f_{GCLK}}$$

$$T_{HIGH} = \frac{BAUD + 5}{f_{GCLK}}$$

**Note:** The $I^2C$ standard *Fm+* (Fast-mode plus) requires a nominal high to low SCL ratio of 1:2, and BAUD should be set accordingly. At a minimum, BAUD.BAUD and/or BAUD.BAUDLOW must be non-zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Startup Timing** The minimum time between SDA transition and SCL rising edge is 6 APB cycles when the DATA register is written in smart mode. If a greater startup time is required due to long rise times, the time between DATA write and IF clear must be controlled by software.
**Note:** When timing is controlled by user, the Smart Mode cannot be enabled.

**Host Clock Generation (High-Speed Mode)**
For I²C *Hs* transfers, there is no SCL synchronization. Instead, the SCL frequency is determined by the GCLK_SERCOMx_CORE frequency ($f_{GCLK}$) and the High-Speed Baud setting in the Baud register (BAUD.HSBAUD). When BAUD.HSBAUDLOW=0, the HSBAUD value will determine both SCL high and SCL low. In this case the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + 2 \cdot HS\,BAUD}$$

When HSBAUDLOW is non-zero, the following formula determines the SCL frequency.

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\,BAUD + HSBAUDLOW}$$

**Note:** The I²C standard *Hs* (High-speed) requires a nominal high to low SCL ratio of 1:2, and HSBAUD should be set accordingly. At a minimum, BAUD.HSBAUD and/or BAUD.HSBAUDLOW must be non-zero.

### 36.6.2.4.2. Transmitting Address Packets

The I²C host starts a bus transaction by writing the I²C client address to ADDR.ADDR and the direction bit, as described in Principle of Operation. If the bus is busy, the I²C host will wait until the bus becomes idle before continuing the operation. When the bus is idle, the I²C host will issue a start condition on the bus. The I²C host will then transmit an address packet using the address written to ADDR.ADDR. After the address packet has been transmitted by the I²C host, one of four cases will arise according to arbitration and transfer direction.

**Case 1: Arbitration lost or bus error during address packet transmission**

If arbitration was lost during transmission of the address packet, the Host on Bus bit in the Interrupt Flag Status and Clear register (INTFLAG.MB) and the Arbitration Lost bit in the Status register (STATUS.ARBLOST) are both set. Serial data output to SDA is disabled, and the SCL is released, which disables clock stretching. In effect the I²C host is no longer allowed to execute any operation on the bus until the bus is idle again. A bus error will behave similarly to the Arbitration Lost condition. In this case, the MB Interrupt flag and Host Bus Error bit in the Status register (STATUS.BUSERR) are both set in addition to STATUS.ARBLOST.

The Host Received Not Acknowledge bit in the Status register (STATUS.RXNACK) will always contain the last successfully received acknowledge or not acknowledge indication.

In this case, software will typically inform the application code of the condition and then clear the Interrupt flag before exiting the interrupt routine. No other flags have to be cleared at this moment, because all flags will be cleared automatically the next time the ADDR.ADDR register is written.

**Case 2: Address packet transmit complete – No ACK received**

If there is no I²C client device responding to the address packet, then the INTFLAG.MB Interrupt flag and STATUS.RXNACK will be set. The clock hold is active at this point, preventing further activity on the bus.

The missing ACK response can indicate that the I²C client is busy with other tasks or sleeping. Therefore, it is not able to respond. In this event, the next step can be either issuing a Stop condition (recommended) or resending the address packet by a repeated Start condition. When using SMBus logic, the client must ACK the address. If there is no response, it means that the client is not available on the bus.

**Case 3: Address packet transmit complete – Write packet, Host on Bus set**


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

If the I²C host receives an acknowledge response from the I²C client, INTFLAG.MB will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Initiate a data transmit operation by writing the data byte to be transmitted into DATA.DATA.
- Transmit a new address packet by writing ADDR.ADDR. A repeated Start condition will automatically be inserted before the address packet.
- Issue a Stop condition, consequently terminating the transaction.

**Case 4: Address packet transmit complete – Read packet, Client on Bus set**

If the I²C host receives an ACK from the I²C client, the I²C host proceeds to receive the next byte of data from the I²C client. When the first data byte is received, the Client on Bus bit in the Interrupt Flag register (INTFLAG.SB) will be set and STATUS.RXNACK will be cleared. The clock hold is active at this point, preventing further activity on the bus.

In this case, the software implementation becomes highly protocol dependent. Three possible actions can enable the I²C operation to continue:

- Let the I²C host continue to read data by acknowledging the data received. ACK can be sent by software, or automatically in Smart mode.
- Transmit a new address packet.
- Terminate the transaction by issuing a Stop condition.

**Note:** An ACK or NACK will be automatically transmitted if Smart mode is enabled. The Acknowledge Action bit in the Control B register (CTRLB.ACKACT) determines whether ACK or NACK should be sent.

#### 36.6.2.4.3. Transmitting Data Packets

When an address packet with direction Host Write (see Figure 36-3) was transmitted successfully , INTFLAG.MB will be set. The I²C host will start transmitting data via the I²C bus by writing to DATA.DATA, and monitor continuously for packet collisions.

If a collision is detected, the I²C host will lose arbitration and STATUS.ARBLOST will be set. If the transmit was successful, the I²C host will receive an ACK bit from the I²C client, and STATUS.RXNACK will be cleared. INTFLAG.MB will be set in both cases, regardless of arbitration outcome.

It is recommended to read STATUS.ARBLOST and handle the arbitration lost condition in the beginning of the I²C Host on Bus interrupt. This can be done as there is no difference between handling address and data packet arbitration.

STATUS.RXNACK must be checked for each data packet transmitted before the next data packet transmission can commence. The I²C host is not allowed to continue transmitting data packets if a NACK is received from the I²C client.

#### 36.6.2.4.4. Receiving Data Packets (SCLSM=0)

When INTFLAG.SB is set, the I²C host will already have received one data packet. The I²C host must respond by sending either an ACK or NACK. Sending a NACK may be unsuccessful when arbitration is lost during the transmission. In this case, a lost arbitration will prevent setting INTFLAG.SB. Instead, INTFLAG.MB will indicate a change in arbitration. Handling of lost arbitration is the same as for data bit transmission.

#### 36.6.2.4.5. Receiving Data Packets (SCLSM=1)

When INTFLAG.SB is set, the I²C host will already have received one data packet and transmitted an ACK or NACK, depending on CTRLB.ACKACT. At this point, CTRLB.ACKACT must be set to the correct value for the next ACK bit, and the transaction can continue by reading DATA and issuing a command if not in the Smart mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

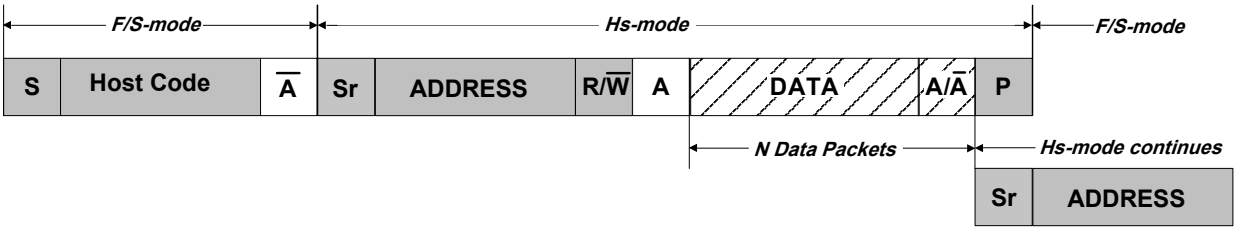
#### 36.6.2.4.6. High-Speed Mode

High-speed transfers are a multi-step process, see High Speed Transfer.

First, a host code (0b00001nnn, where 'nnn' is a unique host code) is transmitted in Full-speed mode, followed by a NACK since no client should acknowledge. Arbitration is performed only during the Full-speed Host Code phase. The host code is transmitted by writing the host code to the Address register (ADDR.ADDR) and writing the High-speed bit (ADDR.HS) to '0'.

After the host code and NACK have been transmitted, the host write interrupt will be asserted. In the meanwhile, the client address can be written to the ADDR.ADDR register together with ADDR.HS=1. Now in High-speed mode, the host will generate a repeated start, followed by the client address with RW-direction. The bus will remain in High-speed mode until a stop is generated. If a repeated start is desired, the ADDR.HS bit must again be written to '1', along with the new address ADDR.ADDR to be transmitted.

**Figure 36-8.** High Speed Transfer



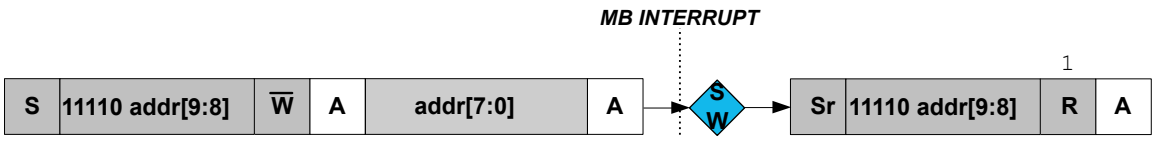

Transmitting in High-speed mode requires the $I^2C$ host to be configured in High-speed mode (CTRLA.SPEED=0x2) and the SCL Clock Stretch mode (CTRLA.SCLSM) bit set to '1'.

#### 36.6.2.4.7. 10-Bit Addressing

When 10-bit addressing is enabled by the Ten Bit Addressing Enable bit in the Address register (ADDR.TENBITEN=1) and the Address bit field ADDR.ADDR is written, the two address bytes will be transmitted, see 10-bit Address Transmission for a Read Transaction. The addressed client acknowledges the two address bytes, and the transaction continues. Regardless of whether the transaction is a read or write, the host must start by sending the 10-bit address with the direction bit (ADDR.ADDR[0]) being zero.

If the host receives a NACK after the first byte, the Write Interrupt flag will be raised and the STATUS.RXNACK bit will be set. If the first byte is acknowledged by one or more clients, then the host will proceed to transmit the second address byte and the host will first see the Write Interrupt flag after the second byte is transmitted. If the transaction direction is read-from-client, the 10-bit address transmission must be followed by a repeated start and the first 7 bits of the address with the read/write bit equal to '1'.

**Figure 36-9.** 10-bit Address Transmission for a Read Transaction




This implies the following procedure for a 10-bit read operation:

1. Write the 10-bit address to ADDR.ADDR[10:1]. ADDR.TENBITEN must be '1', the direction bit (ADDR.ADDR[0]) must be '0' (can be written simultaneously with ADDR).
2. Once the Host on Bus interrupt is asserted, Write ADDR[7:0] register to '11110 `address[9:8]` 1'. ADDR.TENBITEN must be cleared (can be written simultaneously with ADDR).
3. Proceed to transmit data.

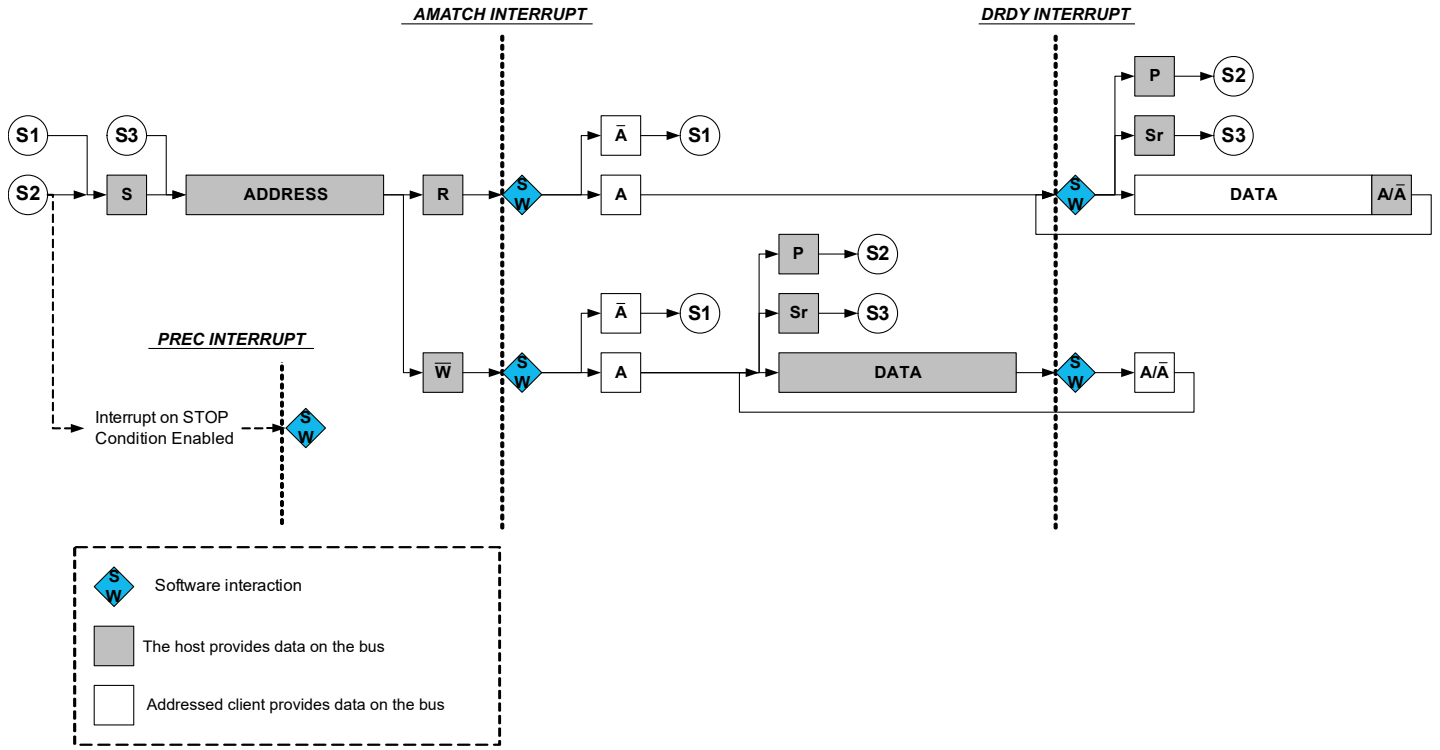

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.6.2.5. I²C Client Operation

The I²C Client is byte-oriented and interrupt-based. The number of interrupts generated is kept at a minimum by automatic handling of most events. The software driver complexity and code size are reduced by auto-triggering of operations, and a special smart mode, which can be enabled by the Smart Mode Enable bit in the Control B register (CTRLB.SMEN).

This diagram is used as reference for the description of the I²C Client operation throughout the document.

**Figure 36-10.** I²C Client Behavioral Diagram




#### Receiving Address Packets

When a start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C Client will wait for a new start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C Client clears INTFLAG.AMATCH. As the I²C Client holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read / Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C Client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C Host, one of two cases will arise based on transfer direction.

**Case 1: Address packet accepted – Read flag set**


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The STATUS.DIR bit is '1', indicating an I²C Host read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C Client hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I²C Client will wait for a new start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I²C Client Command bit field in the Control B register (CTRLB.CMD) can be written to '0x3' for both read and write operations as the command execution is dependent on the STATUS.DIR bit. Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

**Case 2: Address packet accepted – Write flag set**

The STATUS.DIR bit is cleared, indicating an I²C Host write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I²C Client will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C Client will wait for a new start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I²C Client command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

**Receiving and Transmitting Data Packets**

After the I²C Client has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA. When a data packet is received or sent, INTFLAG.DRDY will be set. After receiving data, the I²C Client will send an acknowledge according to CTRLB.ACKACT.

**Case 1: Data received**

INTFLAG.DRDY is set, and SCL is held low, pending for SW interaction.

**Case 2: Data sent**

When a byte transmission is successfully completed, the INTFLAG.DRDY interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I²C Client must expect a stop or a repeated start to be received. The I²C Client must release the data line to allow the I²C Host to generate a stop or repeated start. Upon detecting a stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I²C Client will return to IDLE state.

#### 36.6.2.5.1. Receiving Address Packets (SCLSM=0)

When CTRLA.SCLSM=0, the I²C client stretches the SCL line according to Figure 36-10. When the I²C client is properly configured, it will wait for a Start condition.

When a Start condition is detected, the successive address packet will be received and checked by the address match logic. If the received address is not a match, the packet will be rejected, and the I²C client will wait for a new Start condition. If the received address is a match, the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) will be set.

SCL will be stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client holds the clock by forcing SCL low, the software has unlimited time to respond.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, this indicates that the last packet addressed to the I²C client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. Therefore, the next AMATCH interrupt is the first


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (ARP).

After the address packet has been received from the I²C host, one of two cases will arise based on transfer direction.

**Case 1: Address packet accepted – Read flag set**

The STATUS.DIR bit is '1', indicating an I²C host read operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, I²C client hardware will set the Data Ready bit in the Interrupt Flag register (INTFLAG.DRDY), indicating data are needed for transmit. If a NACK is sent, the I²C client will wait for a new Start condition and address match.

Typically, software will immediately acknowledge the address packet by sending an ACK/NACK bit. The I²C client Command bit field in the Control B register (CTRLB.CMD) can be written to '0x3' for both read and write operations as the command execution is dependent on the STATUS.DIR bit. Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

**Case 2: Address packet accepted – Write flag set**

The STATUS.DIR bit is cleared, indicating an I²C host write operation. The SCL line is forced low, stretching the bus clock. If an ACK is sent, the I²C client will wait for data to be received. Data, repeated start or stop can be received.

If a NACK is sent, the I²C client will wait for a new Start condition and address match. Typically, software will immediately acknowledge the address packet by sending an ACK/NACK. The I²C client command CTRLB.CMD = 3 can be used for both read and write operation as the command execution is dependent on STATUS.DIR.

Writing '1' to INTFLAG.AMATCH will also cause an ACK/NACK to be sent corresponding to the CTRLB.ACKACT bit.

#### 36.6.2.5.2. Receiving Address Packets (SCLSM = 1)

When SCLSM = 1, the I²C client will stretch the SCL line only after an ACK, refer to the "Client Behavioral Diagram (SCLSM = 1)". When the I²C client is properly configured, it will wait for a Start condition to be detected.

When a Start condition is detected, the successive address packet will be received and checked by the address match logic.

If the received address is not a match, the packet will be rejected and the I²C client will wait for a new Start condition.

If the address matches, the acknowledge action as configured by the Acknowledge Action bit Control B register (CTRLB.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register (INTFLAG.AMATCH) is set. SCL will be stretched until the I²C client clears INTFLAG.AMATCH. As the I²C client holds the clock by forcing SCL low, the software is given unlimited time to respond to the address.

The direction of a transaction is determined by reading the Read/Write Direction bit in the Status register (STATUS.DIR). This bit will be updated only when a valid address packet is received.

If the Transmit Collision bit in the Status register (STATUS.COLL) is set, the last packet addressed to the I²C client had a packet collision. A collision causes the SDA and SCL lines to be released without any notification to software. The next AMATCH interrupt is, therefore, the first indication of the previous packet's collision. Collisions are intended to follow the SMBus Address Resolution Protocol (*ARP*).

After the address packet has been received from the I²C host, INTFLAG.AMATCH be set to '1' to clear it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 36.6.2.5.3. Receiving and Transmitting Data Packets

After the I²C client has received an address packet, it will respond according to the direction either by waiting for the data packet to be received or by starting to send a data packet by writing to DATA.DATA. When a data packet is received or sent, INTFLAG.DRDY will be set. After receiving data, the I²C client will send an acknowledge according to CTRLB.ACKACT.

**Case 1: Data received**

INTFLAG.DRDY is set, and SCL is held low, pending for SW interaction.

**Case 2: Data sent**

When a byte transmission is successfully completed, the INTFLAG.DRDY Interrupt flag is set. If NACK is received, indicated by STATUS.RXNACK=1, the I²C client must expect a stop or a repeated start to be received. The I²C client must release the data line to allow the I²C host to generate a stop or repeated start. Upon detecting a Stop condition, the Stop Received bit in the Interrupt Flag register (INTFLAG.PREC) will be set and the I²C client will return to IDLE state.

#### 36.6.2.5.4. High-Speed Mode

When the I²C client is configured in High-speed mode (*Hs*, CTRLA.SPEED=0x2) and CTRLA.SCLSM=1, switching between Full-speed and High-speed modes is automatic. When the client recognizes a START followed by a host code transmission and a NACK, it automatically switches to High-speed mode and sets the High-speed status bit (STATUS.HS). The client will then remain in High-speed mode until a STOP is received.

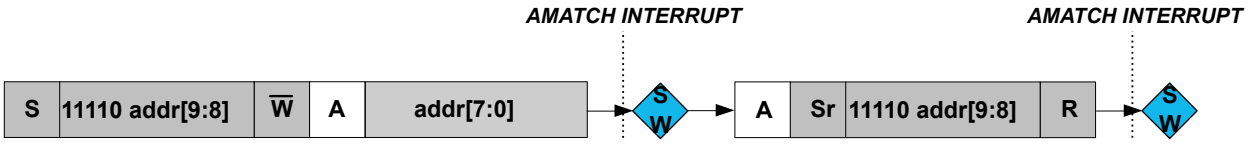
#### 36.6.2.5.5. 10-Bit Addressing

When 10-bit addressing is enabled (ADDR.TENBITEN=1), the two address bytes following a START will be checked against the 10-bit client address recognition. The first byte of the address will always be acknowledged, and the second byte will raise the address Interrupt flag, see 10-bit Addressing.

If the transaction is a write, then the 10-bit address will be followed by *N* data bytes.

If the operation is a read, the 10-bit address will be followed by a repeated START and reception of '11110 `ADDR[9:8]` 1', and the second address interrupt will be received with the DIR bit set. The client matches on the second address as it was addressed by the previous 10-bit address.

**Figure 36-11.** 10-bit Addressing




#### 36.6.2.5.6. PMBus Group Command

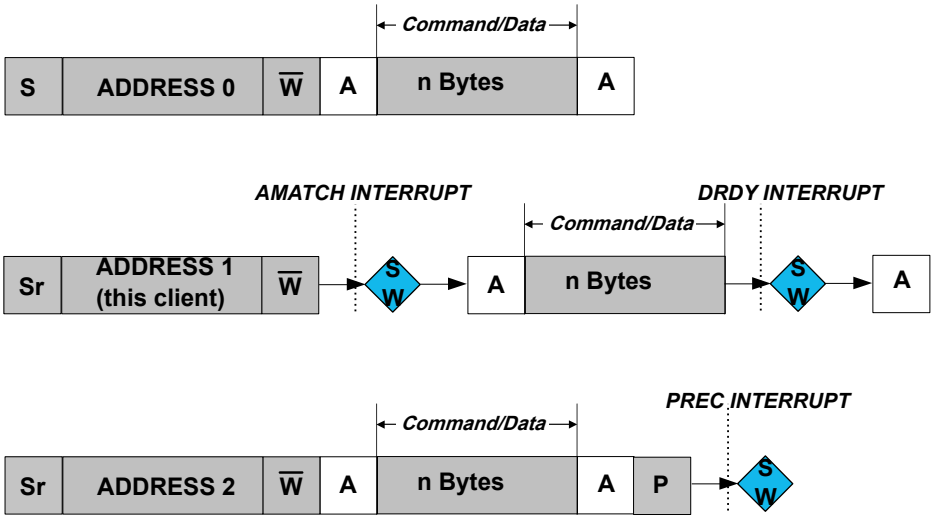
When the PMBus Group Command bit in the CTRLB register is set (CTRLB.GCMD=1) and 7-bit addressing is used, INTFLAG.PREC will be set if the client has been addressed since the last STOP condition. When CTRLB.GCMD=0, a STOP condition without address match will not be set INTFLAG.PREC.

The group command protocol is used to send commands to more than one device. The commands are sent in one continuous transmission with a single STOP condition at the end. When the STOP condition is detected by the clients addressed during the group command, they all begin executing the command they received.

The following figure shows an example where this client, bearing ADDRESS 1, is addressed after a repeated START condition. There can be multiple clients addressed before and after this client. Eventually, at the end of the group command, a single STOP is generated by the host. At this point a STOP interrupt is asserted.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 36-12.** PMBus Group Command Example




## 36.6.3. Additional Features

### 36.6.3.1. SMBus

The I²C includes three hardware SCL low time-outs which allow a time-out to occur for SMBus SCL low time-out, host extend time-out, and client extend time-out. This allows for SMBus functionality These time-outs are driven by the GCLK_SERCOM_SLOW clock. The GCLK_SERCOM_SLOW clock is used to accurately time the time-out and must be configured to use a 32.768 kHz oscillator. The I²C interface also allows for a SMBus compatible SDA hold time.

- **$T_{TIMEOUT}$:** SCL low time of 25..35ms – Measured for a single SCL low period. It is enabled by CTRLA.LOWTOUTEN
- **$T_{LOW:SEXT}$:** Cumulative clock low extend time of 25 ms – Measured as the cumulative SCL low extend time by a client device in a single message from the initial START to the STOP. It is enabled by CTRLA.SEXTTOEN.
- **$T_{LOW:MEXT}$:** Cumulative clock low extend time of 10 ms – Measured as the cumulative SCL low extend time by the host device within a single byte from START-to-ACK, ACK-to-ACK, or ACK-to-STOP. It is enabled by CTRLA.MEXTTOEN.

### 36.6.3.2. Smart Mode

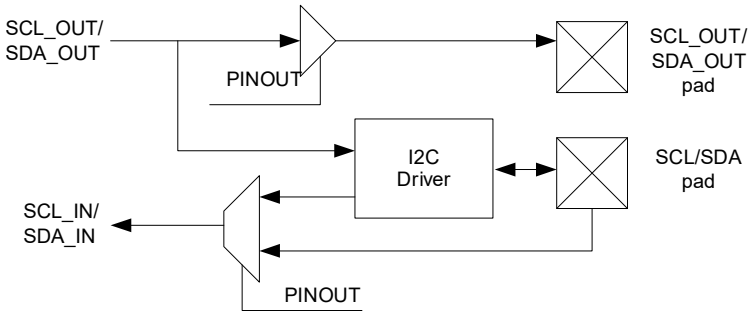
The I²C interface has a Smart mode that simplifies application code and minimizes the user interaction needed to adhere to the I²C protocol. The Smart mode accomplishes this by automatically issuing an ACK or NACK (based on the content of CTRLB.ACKACT) as soon as DATA.DATA is read.

### 36.6.3.3. 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control A register (CTRLA.PINOUT) will enable 4-Wire mode operation. In this mode, the internal I²C tri-state drivers are bypassed, and an external I²C compliant tri-state driver is needed when connecting to an I²C bus.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 36-13.** I²C Pad Interface




### 36.6.3.4. Quick Command

Setting the Quick Command Enable bit in the Control B register (CTRLB.QCEN) enables quick command. When quick command is enabled, the corresponding Interrupt flag (INTFLAG.SB or INTFLAG.MB) is set immediately after the client acknowledges the address. At this point, the software can either issue a Stop command or a repeated start by writing CTRLB.CMD or ADDR.ADDR.

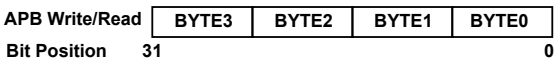
### 36.6.3.5. 32-bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled by writing to the Data 32-bit bit field in the Control C register (CTRLC.DATA32B=1). When enabled, write and read transaction to/from the DATA register are 32 bit in size.

If frames are not multiples of 4 Bytes, the Length Counter (LENGTH.LEN) and Length Enable (LENGTH.LENEN) must be configured before data transfer begins. LENGTH.LEN must be enabled only when CTRLC.DATA32B is enabled.

The figure below shows the order of transmit and receive when using 32-bit mode. Bytes are transmitted or received and stored in order from 0 to 3.

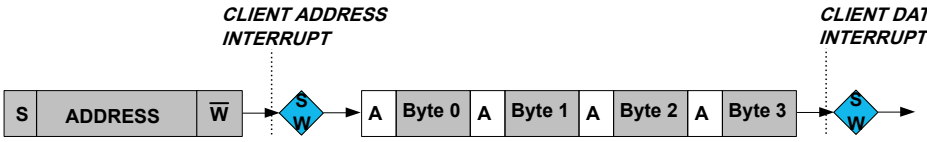
**Figure 36-14.** 32-bit Extension Byte Ordering




#### 32-bit Extension Client Operation

The figure below shows a transaction with 32-bit Extension enabled (CTRLC.DATA32B=1). In Client operation, the Address Match interrupt in the Interrupt Flag Status and Clear register (INTFLAG.AMATCH) is set after the address is received and available in the DATA register. The Data Ready interrupt (INTFLAG.DRDY) will then be raised for every 4 Bytes transferred.

**Figure 36-15.** 32-bit Extension Client Operation




The LENGTH register can be written before the frame begins, or when the AMATCH interrupt is set. If the frame size is not LENGTH.LEN Bytes, the Length Error status bit (STATUS.LENERR) is raised. If LENGTH.LEN is not a multiple of 4 Bytes, the final INTFLAG.DRDY interrupt is raised when the last Byte is received for Host reads. For Host writes, the last data byte will be automatically NACKed. On address recognition, the internal length counter is reset in preparation for the incoming frame.

High Speed transactions start with a Full Speed Host Code. When a Host Code is detected, no data is received and the next expected operation is a repeated start. For this reason, the length is not


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

counted after a Host Code is received. In this case, no Length Error (STATUS.LENERR) is registered, regardless of the LENGTH.LENEN setting.

When SCL clock stretch mode is selected (CTRLA.SCLSM=1) and the transaction is a Host write, the selected Acknowledge Action (CTRLB.ACKACT) will only be used to ACK/NACK each 4th byte. All other bytes are ACKed. This allows the user to write CTRLB.ACKACT=1 in the final interrupt, so that the last byte in a 32-bit word will be NACKed.

Writing to the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 Bytes, the remainder will be lost.

#### 32-bit Extension Host Operation

When using the $I^2C$ configured as Host, the Address register must be written with the desired address (ADDR.ADDR), and optionally, the transaction Length and transaction Length Enable bits (ADDR.LEN and ADDR.LENEN) can be written. When ADDR.LENEN is written to '1' along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. Then, the ADDR.LEN bytes are transferred, followed by an automatically generated NACK (for Host reads) and a STOP.

The INTFLAG.SB or INTFLAG.MB are raised for every 4 Bytes transferred. If the transaction is a Host read and ADDR.LEN is not a multiple of 4 Bytes, the final INTFLAG.SB is set when the last byte is received.

When SCL clock stretch mode is enabled (CTRLA.SCLSM=1) and the transaction is a Host read, the selected Acknowledge Action (CTRLB.ACKACT) will only be used to ACK/NACK each 4th Byte. All other bytes are ACKed. This allows the user to set CTRLB.ACKACT=1 in the final interrupt, so that the last byte in a 32-bit word will be NACKed.

If a NACK is received by the Client for a Host write transaction before ADDR.LEN bytes, a STOP will be automatically generated, and the length error (STATUS.LENERR) is raised along with the INTFLAG.ERROR interrupt.

### 36.6.4. DMA, Interrupts and Events

Each interrupt source has its own interrupt flag. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request is active until the interrupt flag is cleared, the interrupt is disabled or the $I^2C$ is reset. See the INTFLAG (Client) or INTFLAG (Host) register for details on how to clear interrupt flags.

**Table 36-1.** Module Request for SERCOM $I^2C$ Client

| Condition | Request | | |
|---|---|---|---|
| | DMA | Interrupt | Event |
| Data needed for transmit (TX) (Client transmit mode) | Yes (request cleared when data is written) | | NA |
| Data received (RX) (Client receive mode) | Yes (request cleared when data is read) | | |
| Data Ready (DRDY) | | Yes | |
| Address Match (AMATCH) | | Yes | |
| Stop received (PREC) | | Yes | |
| Error (ERROR) | | Yes | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 36-2.** Module Request for SERCOM I²C Host

| Condition | Request | | |
|---|---|---|---|
| | DMA | Interrupt | Event |
| Data needed for transmit (TX) (Host transmit mode) | Yes (request cleared when data is written) | | NA |
| Data needed for transmit (RX) (Host transmit mode) | Yes (request cleared when data is read) | | |
| Host on Bus (MB) | | Yes | |
| Stop received (SB) | | Yes | |
| Error (ERROR) | | Yes | |

#### 36.6.4.1. DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=1.

##### 36.6.4.1.1. Client DMA

When using the I²C client with DMA, an address match will cause the address Interrupt flag (INTFLAG.ADDRMATCH) to be raised. After the interrupt has been serviced, data transfer will be performed through DMA.

The I²C client generates the following requests:

- Write data received (RX): The request is set when host write data is received. The request is cleared when DATA is read.
- Read data needed for transmit (TX): The request is set when data is needed for a host read operation. The request is cleared when DATA is written.

##### 36.6.4.1.2. Host DMA

When using the I²C host with DMA, the ADDR register must be written with the desired address (ADDR.ADDR), transaction length (ADDR.LEN), and transaction length enable (ADDR.LENEN). When ADDR.LENEN is written to 1 along with ADDR.ADDR, ADDR.LEN determines the number of data bytes in the transaction from 0 to 255. DMA is then used to transfer ADDR.LEN bytes followed by an automatically generated NACK (for host reads) and a STOP.

If a NACK is received by the client for a host write transaction before ADDR.LEN bytes, a STOP will be automatically generated and the length error (STATUS.LENERR) will be raised along with the INTFLAG.ERROR interrupt.

The I²C host generates the following requests:

- Read data received (RX): The request is set when host read data is received. The request is cleared when DATA is read.
- Write data needed for transmit (TX): The request is set when data is needed for a host write operation. The request is cleared when DATA is written.

#### 36.6.4.2. Interrupts

The I²C Client has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The I²C Host has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Error (ERROR)
- Client on Bus (SB)
- Host on Bus (MB)

Each interrupt source has its own Interrupt flag. The Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) will be set when the Interrupt condition is meet. Each interrupt can be individually enabled by writing '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR). An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request active until the Interrupt flag is cleared, the interrupt is disabled or the $I^2C$ is reset. See the INTFLAG register for details on how to clear Interrupt flags.

The value of INTFLAG indicates which interrupt is executed. Note that interrupts must be globally enabled for interrupt requests. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller

#### 36.6.4.3. Events

Not applicable.

### 36.6.5. Sleep Mode Operation

**$I^2C$ Host Operation**

The generic clock (GCLK_SERCOMx_CORE) will continue to run in idle sleep mode. If the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY) is '1', the GLK_SERCOMx_CORE will also run in Standby Sleep mode. Any interrupt can wake-up the device.

If CTRLA.RUNSTDBY=0, the GLK_SERCOMx_CORE will be disabled after any ongoing transaction is finished. Any interrupt can wake-up the device.

**$I^2C$ Client Operation**

Writing CTRLA.RUNSTDBY=1 will allow the Address Match interrupt to wake-up the device.

When CTRLA.RUNSTDBY=0, all receptions will be dropped.

### 36.6.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the CTRLA register (CTRLA.SWRST)
- Enable bit in the CTRLA register (CTRLA.ENABLE)
- Command bits in CTRLB register (CTRLB.CMD)
- Write to Bus State bits in the Status register (STATUS.BUSSTATE)
- Address bits in the Address register (ADDR.ADDR) when in host operation.

The following registers are synchronized when written:

- Data (DATA) when in host operation
- Length (LENGTH) when in client operation

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 36.7. Register Summary - I2C Client

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | SEXTTOEN | | SDAHOLD[1:0] | | | | | PINOUT |
| | | 31:24 | | LOWTOUTEN | | | SCLSM | | SPEED[1:0] | |
| 0x04 | CTRLB | 7:0 | | | | | | | | |
| | | 15:8 | AMODE[1:0] | | | | | AACKEN | GCMD | SMEN |
| | | 23:16 | | | | | | ACKACT | CMD[1:0] | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | | | | | SDASETUP[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | DATA32B |
| 0x0C ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | ERROR | | | | | DRDY | AMATCH | PREC |
| 0x15 | Reserved | | | | | | | | | |
| 0x16 | INTENSET | 7:0 | ERROR | | | | | DRDY | AMATCH | PREC |
| 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTFLAG | 7:0 | ERROR | | | | | DRDY | AMATCH | PREC |
| 0x19 | Reserved | | | | | | | | | |
| 0x1A | STATUS | 7:0 | CLKHOLD | LOWTOUT | | SR | DIR | RXNACK | COLL | BUSERR |
| | | 15:8 | | | | | LENERR | HS | SEXTTOUT | |
| 0x1C | SYNCBUSY | 7:0 | | | | LENGTH | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 ... 0x21 | Reserved | | | | | | | | | |
| 0x22 | LENGTH | 7:0 | LEN[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | LENEN |
| 0x24 | ADDR | 7:0 | ADDR[6:0] | | | | | | | GENCEN |
| | | 15:8 | TENBITEN | | | | | ADDR[9:7] | | |
| | | 23:16 | ADDRMASK[6:0] | | | | | | | |
| | | 31:24 | | | | | | ADDRMASK[9:7] | | |
| 0x28 | DATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |

## 36.8. Register Description - I²C Client

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | LOWTOUTEN | | | SCLSM | | SPEED[1:0] | |
| Access | | R/W | | | R/W | | R/W | R/W |
| Reset | | 0 | | | 0 | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SEXTTOEN | | SDAHOLD[1:0] | | | | | PINOUT |
| Access | R/W | | R/W | R/W | | | | R/W |
| Reset | 0 | | 0 | 0 | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 30 – LOWTOUTEN** SCL Low Time-Out
This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the client will release its clock hold, if enabled, and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Time-out disabled. |
| 1 | Time-out enabled. |

**Bit 27 – SCLSM** SCL Clock Stretch Mode
This bit controls when SCL will be stretched for software interaction.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | SCL stretch according to Figure 36-10 |
| 1 | SCL stretch only after ACK bit according to Figure 36-10 |

**Bits 25:24 – SPEED[1:0]** Transfer Speed
These bits define bus speed.
These bits are not synchronized.

| Value | Description |
|---|---|
| 0x0 | Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz |
| 0x1 | Fast-mode Plus (Fm+) up to 1 MHz |
| 0x2 | High-speed mode (Hs-mode) up to 3.4 MHz |
| 0x3 | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 23 – SEXTTOEN** Client SCL Low Extend Time-Out

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the client will release its clock hold if enabled and reset the internal state machine. Any interrupt flags set at the time of time-out will remain set. If the address was recognized, PREC will be set when a STOP is received.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Time-out disabled |
| 1 | Time-out enabled |

**Bits 21:20 – SDAHOLD[1:0]** SDA Hold Time

These bits define the SDA hold time with respect to the negative edge of SCL.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIS | Disabled |
| 0x1 | 75 | 50-100ns hold time |
| 0x2 | 450 | 300-600ns hold time |
| 0x3 | 600 | 400-800ns hold time |

**Bit 16 – PINOUT** Pin Usage

This bit sets the pin usage to either two- or four-wire operation:
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | 4-wire operation disabled |
| 1 | 4-wire operation enabled |

**Bit 7 – RUNSTDBY** Run in Standby

This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Disabled – All reception is dropped. |
| 1 | Wake on address match, if enabled. |

**Bits 4:2 – MODE[2:0]** Operating Mode

These bits must be written to 0x04 to select the $I^2C$ client serial communication interface of the SERCOM.
These bits are not synchronized.

**Bit 1 – ENABLE** Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled or being disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ACKACT | CMD[1:0] | |
| Access | | | | | | R/W | W | W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | AMODE[1:0] | | | | | AACKEN | GCMD | SMEN |
| Access | R/W | R/W | | | | R/W | R/W | R/W |
| Reset | 0 | 0 | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bit 18 – ACKACT** Acknowledge Action

This bit defines the client's acknowledge behavior after an address or data byte is received from the host. The acknowledge action is executed when a command is written to the CMD bits. If smart mode is enabled (CTRLB.SMEN=1), the acknowledge action is performed when the DATA register is read.
ACKACT shall not be updated more than once between each peripheral interrupts request.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | Send ACK |
| 1 | Send NACK |

**Bits 17:16 – CMD[1:0]** Command

This bit field triggers the client operation as the below. The CMD bits are strobe bits, and always read as zero. The operation is dependent on the client interrupt flags, INTFLAG.DRDY and INTFLAG.AMATCH, in addition to STATUS.DIR.
All interrupt flags (INTFLAG.DRDY, INTFLAG.AMATCH and INTFLAG.PREC) are automatically cleared when a command is given.
This bit is not enable-protected.

**Table 36-3.** Command Description

| CMD[1:0] | DIR | Action |
|---|---|---|
| 0x0 | X | (No action) |
| 0x1 | X | (Reserved) |
| 0x2 | Used to complete a transaction in response to a data interrupt (DRDY) | |
| | 0 (Host write) | Execute acknowledge action succeeded by waiting for any start (S/Sr) condition |
| | 1 (Host read) | Wait for any start (S/Sr) condition |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 36-3.** Command Description (continued)

| CMD[1:0] | DIR | Action |
|---|---|---|
| 0x3 | Used in response to an address interrupt (AMATCH) | |
| | 0 (Host write) | Execute acknowledge action succeeded by reception of next byte |
| | 1 (Host read) | Execute acknowledge action succeeded by client data interrupt |
| | Used in response to a data interrupt (DRDY) | |
| | 0 (Host write) | Execute acknowledge action succeeded by reception of next byte |
| | 1 (Host read) | Execute a byte read operation followed by ACK/NACK reception |

**Bits 15:14 – AMODE[1:0]** Address Mode

These bits set the addressing mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | MASK | The client responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK. See *SERCOM – Serial Communication Interface* for additional information. |
| 0x1 | 2_ADDRS | The client responds to the two unique addresses in ADDR.ADDR and ADDR.ADDRMASK. |
| 0x2 | RANGE | The client responds to the range of addresses between and including ADDR.ADDR and ADDR.ADDRMASK. ADDR.ADDR is the upper limit. |
| 0x3 | - | Reserved. |

**Bit 10 – AACKEN** Automatic Acknowledge Enable

This bit enables the address to be automatically acknowledged if there is an address match.

| Value | Description |
|---|---|
| 0 | Automatic acknowledge is disabled. |
| 1 | Automatic acknowledge is enabled. |

**Bit 9 – GCMD** PMBus Group Command

This bit enables PMBus group command support. When enabled, the Stop Received interrupt flag (INTFLAG.PREC) will be set when a STOP condition is detected if the client has been addressed since the last STOP condition on the bus.

| Value | Description |
|---|---|
| 0 | Group command is disabled. |
| 1 | Group command is enabled. |

**Bit 8 – SMEN** Smart Mode Enable

When smart mode is enabled, data is acknowledged automatically when DATA.DATA is read.

| Value | Description |
|---|---|
| 0 | Smart mode is disabled. |
| 1 | Smart mode is enabled. |

**Related Links**

SERCOM – Serial Communication Interface


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DATA32B |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SDASETUP[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 24 – DATA32B** Data 32 Bit
This bit enables 32-bit data writes and reads to/from the DATA register.

| Value | Description |
|---|---|
| 0 | Data transaction to/from DATA are 8-bit in size |
| 1 | Data transaction to/from DATA are 32-bit in size |

**Bits 3:0 – SDASETUP[3:0]** SDA Setup Time
These bits select the minimum SDA-to-SCL setup time, measured from the release of SDA to the release of SCL:
$t_{SU:DAT}$ = (CLK_SERCOMx_APB period) x (6 + 16 x SDASETUP)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | | DRDY | AMATCH | PREC |
| Access | R/W | | | | | R/W | R/W | R/W |
| Reset | 0 | | | | | 0 | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 2 – DRDY** Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Ready bit, which disables the Data Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Data Ready interrupt is disabled. |
| 1 | The Data Ready interrupt is enabled. |

**Bit 1 – AMATCH** Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Address Match Interrupt Enable bit, which disables the Address Match interrupt.

| Value | Description |
|---|---|
| 0 | The Address Match interrupt is disabled. |
| 1 | The Address Match interrupt is enabled. |

**Bit 0 – PREC** Stop Received Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.

| Value | Description |
|---|---|
| 0 | The Stop Received interrupt is disabled. |
| 1 | The Stop Received interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x16
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | | DRDY | AMATCH | PREC |
| Access | R/W | | | | | R/W | R/W | R/W |
| Reset | 0 | | | | | 0 | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 2 – DRDY** Data Ready Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Data Ready bit, which enables the Data Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Data Ready interrupt is disabled. |
| 1 | The Data Ready interrupt is enabled. |

**Bit 1 – AMATCH** Address Match Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Address Match Interrupt Enable bit, which enables the Address Match interrupt.

| Value | Description |
|---|---|
| 0 | The Address Match interrupt is disabled. |
| 1 | The Address Match interrupt is enabled. |

**Bit 0 – PREC** Stop Received Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Stop Received Interrupt Enable bit, which enables the Stop Received interrupt.

| Value | Description |
|---|---|
| 0 | The Stop Received interrupt is disabled. |
| 1 | The Stop Received interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | | DRDY | AMATCH | PREC |
| Access | R/W | | | | | R/W | R/W | R/W |
| Reset | 0 | | | | | 0 | 0 | 0 |

**Bit 7 – ERROR** Error
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register. The corresponding bits in STATUS are LENERR, SEXTTOUT, LOWTOUT, COLL, and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 2 – DRDY** Data Ready
This flag is set when a $I^2C$ client byte transmission or reception is successfully completed.
The flag is cleared by hardware when either:

- Writing to the DATA register.
- Reading the DATA register with smart mode enabled.
- Writing a valid command to the CMD register.

Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Data Ready interrupt flag.

**Bit 1 – AMATCH** Address Match
This flag is set when the $I^2C$ client address match logic detects that a valid address has been received.
The flag is cleared by hardware when CTRL.CMD is written.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Address Match interrupt flag. When cleared, an ACK/NACK will be sent according to CTRLB.ACKACT.

**Bit 0 – PREC** Stop Received
This flag is set when a stop condition is detected for a transaction being processed. A stop condition detected between a bus host and another client will not set this flag, unless the PMBus Group Command is enabled in the Control B register (CTRLB.GCMD=1).
This flag is cleared by hardware after a command is issued on the next address match.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Stop Received interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.7. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LENERR | HS | SEXTTOUT | |
| Access | | | | | R/W | R/W | R/W | |
| Reset | | | | | 0 | 0 | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CLKHOLD | LOWTOUT | | SR | DIR | RXNACK | COLL | BUSERR |
| Access | R | R/W | | R | R | R | R/W | R/W |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 0 |

**Bit 11 – LENERR** Transaction Length Error
This bit is set when the length counter is enabled (LENGTH.LENEN) and a STOP or repeated START is received before or after the length in LENGTH.LEN is reached.
This bit is cleared automatically when responding to a new start condition with ACK or NACK (CTRLB.CMD=0x3) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

**Bit 10 – HS** High-speed
This bit is set if the client detects a START followed by a Host Code transmission.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status. However, this flag is automatically cleared when a STOP is received.

**Bit 9 – SEXTTOUT** Client SCL Low Extend Time-Out
This bit is set if a client SCL low extend time-out occurs.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

| Value | Description |
|---|---|
| 0 | No SCL low extend time-out has occurred. |
| 1 | SCL low extend time-out has occurred. |

**Bit 7 – CLKHOLD** Clock Hold
The client Clock Hold bit (STATUS.CLKHOLD) is set when the client is holding the SCL line low, stretching the $I^2C$ clock. Software should consider this bit a read-only status flag that is set when INTFLAG.DRDY or INTFLAG.AMATCH is set.
This bit is automatically cleared when the corresponding interrupt is also cleared.

**Bit 6 – LOWTOUT** SCL Low Time-out
This bit is set if an SCL low time-out occurs.
This bit is cleared automatically if responding to a new start condition with ACK or NACK (write 3 to CTRLB.CMD) or when INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | No SCL low time-out has occurred. |
| 1 | SCL low time-out has occurred. |

**Bit 4 – SR** Repeated Start

When INTFLAG.AMATCH is raised due to an address match, SR indicates a repeated start or start condition.
This flag is only valid while the INTFLAG.AMATCH flag is one.

| Value | Description |
|---|---|
| 0 | Start condition on last address match |
| 1 | Repeated start condition on last address match |

**Bit 3 – DIR** Read / Write Direction

The Read/Write Direction (STATUS.DIR) bit stores the direction of the last address packet received from a host.

| Value | Description |
|---|---|
| 0 | Host write operation is in progress. |
| 1 | Host read operation is in progress. |

**Bit 2 – RXNACK** Received Not Acknowledge

This bit indicates whether the last data packet sent was acknowledged or not.

| Value | Description |
|---|---|
| 0 | Host responded with ACK. |
| 1 | Host responded with NACK. |

**Bit 1 – COLL** Transmit Collision

If set, the $I^2C$ client was not able to transmit a high data or NACK bit, the $I^2C$ client will immediately release the SDA and SCL lines and wait for the next packet addressed to it.
This flag is intended for the SMBus address resolution protocol (ARP). A detected collision in non-ARP situations indicates that there has been a protocol violation, and should be treated as a bus error.
Note that this status will not trigger any interrupt, and should be checked by software to verify that the data were sent correctly. This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD), or INTFLAG.AMATCH is cleared.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the status.

| Value | Description |
|---|---|
| 0 | No collision detected on last data byte sent. |
| 1 | Collision detected on last data byte sent. |

**Bit 0 – BUSERR** Bus Error

The Bus Error bit (STATUS.BUSERR) indicates that an illegal bus condition has occurred on the bus, regardless of bus ownership. An illegal bus condition is detected if a protocol violating start, repeated start or stop is detected on the $I^2C$ bus lines. A start condition directly followed by a stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set STATUS.BUSERR.
This bit is cleared automatically if responding to an address match with an ACK or a NACK (writing 0x3 to CTRLB.CMD) or INTFLAG.AMATCH is cleared.
Writing a '1' to this bit will clear the status.
Writing a '0' to this bit has no effect.

| Value | Description |
|---|---|
| 0 | No bus error detected. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Bus error detected. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.8. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | LENGTH | | | ENABLE | SWRST |
| Access | | | | R | | | R | R |
| Reset | | | | 0 | | | 0 | 0 |

**Bit 4 – LENGTH** LENGTH Synchronization Busy
Writing LENGTH requires synchronization. When written, this bit will be set until synchronization is complete. If LENGTH is written while SYNCBUSY.LENGTH is asserted, an APB error will be generated.
**Note:** In client mode, the clock is only running during data transfer, so SYNCBUSY.LENGTH will remain asserted until the next data transfer begins.

| Value | Description |
|---|---|
| 0 | LENGTH synchronization is not busy. |
| 1 | LENGTH synchronization is busy. |

**Bit 1 – ENABLE** SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

| Value | Description |
|---|---|
| 0 | Enable synchronization is not busy. |
| 1 | Enable synchronization is busy. |

**Bit 0 – SWRST** Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

| Value | Description |
|---|---|
| 0 | SWRST synchronization is not busy. |
| 1 | SWRST synchronization is busy. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.9. Length

**Name:** LENGTH
**Offset:** 0x22
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | LENEN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LEN[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 8 – LENEN** Data Length Enable
In 32-bit Extension mode (CTRLC.DATA32B=1), this bit field enables the length counter.

| Value | Description |
|---|---|
| 0 | Length counter is disabled. |
| 1 | Length counter is enabled. |

**Bits 7:0 – LEN[7:0]** Data Length
In 32-bit Extension mode (CTRLC.DATA32B=1) with Data Length counting enabled (LENGTH.LENEN), this bit field configures the data length from 0 to 255 Bytes after which the flag INTFLAG.DRDY is raised.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.10. Address

**Name:** ADDR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ADDRMASK[9:7] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDRMASK[6:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TENBITEN | | | | | ADDR[9:7] | | |
| Access | R/W | | | | | R/W | R/W | R/W |
| Reset | 0 | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[6:0] | | | | | | | GENCEN |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 26:17 – ADDRMASK[9:0]** Address Mask
These bits act as a second address match register, an address mask register or the lower limit of an address range, depending on the CTRLB.AMODE setting.

**Bit 15 – TENBITEN** Ten Bit Addressing Enable

| Value | Description |
|---|---|
| 0 | 10-bit address recognition disabled. |
| 1 | 10-bit address recognition enabled. |

**Bits 10:1 – ADDR[9:0]** Address
These bits contain the $I^2C$ client address used by the client address match logic to determine if a host has addressed the client.
When using 7-bit addressing, the client address is represented by ADDR[6:0].
When using 10-bit addressing (ADDR.TENBITEN=1), the client address is represented by ADDR[9:0]
When the address match logic detects a match, INTFLAG.AMATCH is set and STATUS.DIR is updated to indicate whether it is a read or a write transaction.

**Bit 0 – GENCEN** General Call Address Enable
A general call address is an address consisting of all-zeroes, including the direction bit (host write).

| Value | Description |
|---|---|
| 0 | General call address recognition disabled. |
| 1 | General call address recognition enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.8.11. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Data

The Client data register I/O location (DATA.DATA) provides access to the Host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the Client (STATUS.CLKHOLD is set). An exception occurs when reading the last data byte after the stop condition has been received.

Accessing DATA.DATA auto-triggers $I^2C$ bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 36.9. Register Summary - I2C Host

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | SEXTTOEN | MEXTTOEN | SDAHOLD[1:0] | | | | | PINOUT |
| | | 31:24 | | LOWTOUTEN | INACTOUT[1:0] | | SCLSM | | SPEED[1:0] | |
| 0x04 | CTRLB | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | QCEN | SMEN |
| | | 23:16 | | | | | | ACKACT | CMD[1:0] | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | DATA32B |
| 0x0C | BAUD | 7:0 | BAUD[7:0] | | | | | | | |
| | | 15:8 | BAUDLOW[7:0] | | | | | | | |
| | | 23:16 | HSBAUD[7:0] | | | | | | | |
| | | 31:24 | HSBAUDLOW[7:0] | | | | | | | |
| 0x10 ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | ERROR | | | | | | SB | MB |
| 0x15 | Reserved | | | | | | | | | |
| 0x16 | INTENSET | 7:0 | ERROR | | | | | | SB | MB |
| 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTFLAG | 7:0 | ERROR | | | | | | SB | MB |
| 0x19 | Reserved | | | | | | | | | |
| 0x1A | STATUS | 7:0 | CLKHOLD | LOWTOUT | BUSSTATE[1:0] | | | RXNACK | ARBLOST | BUSERR |
| | | 15:8 | | | | | | LENERR | SEXTTOUT | MEXTTOUT |
| 0x1C | SYNCBUSY | 7:0 | | | | | | SYSOP | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 ... 0x23 | Reserved | | | | | | | | | |
| 0x24 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | TENBITEN | HS | LENEN | | | ADDR[10:8] | | |
| | | 23:16 | LEN[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | DATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x2C ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DBGCTRL | 7:0 | | | | | | | | DBGSTOP |

## 36.10. Register Description - I²C Host

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 36.10.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | LOWTOUTEN | INACTOUT[1:0] | | SCLSM | | SPEED[1:0] | |
| Access | | R/W | R/W | R/W | R/W | | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SEXTTOEN | MEXTTOEN | SDAHOLD[1:0] | | | | | PINOUT |
| Access | R/W | R/W | R/W | R/W | | | | R/W |
| Reset | 0 | 0 | 0 | 0 | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 30 – LOWTOUTEN** SCL Low Time-Out

This bit enables the SCL low time-out. If SCL is held low for 25ms-35ms, the host will release its clock hold, if enabled, and complete the current transaction. A stop condition will automatically be transmitted.
INTFLAG.SB or INTFLAG.MB will be set as normal, but the clock hold will be released. The STATUS.LOWTOUT and STATUS.BUSERR status bits will be set.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Time-out disabled. |
| 1 | Time-out enabled. |

**Bits 29:28 – INACTOUT[1:0]** Inactive Time-Out

If the inactive bus time-out is enabled and the bus is inactive for longer than the time-out setting, the bus state logic will be set to idle. An inactive bus arise when either an $I^2C$ host or client is holding the SCL low.
Enabling this option is necessary for SMBus compatibility, but can also be used in a non-SMBus set-up.
Calculated time-out periods are based on a 100kHz baud rate.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIS | Disabled |
| 0x1 | 55US | 5-6 SCL cycle time-out (50-60µs) |
| 0x2 | 105US | 10-11 SCL cycle time-out (100-110µs) |
| 0x3 | 205US | 20-21 SCL cycle time-out (200-210µs) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 27 – SCLSM** SCL Clock Stretch Mode

This bit controls when SCL will be stretched for software interaction.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | SCL stretch according to Figure 36-5. |
| 1 | SCL stretch only after ACK bit, Figure 36-6. |

**Bits 25:24 – SPEED[1:0]** Transfer Speed

These bits define bus speed.
These bits are not synchronized.

| Value | Description |
|---|---|
| 0x0 | Standard-mode (Sm) up to 100 kHz and Fast-mode (Fm) up to 400 kHz |
| 0x1 | Fast-mode Plus (Fm+) up to 1 MHz |
| 0x2 | High-speed mode (Hs-mode) up to 3.4 MHz |
| 0x3 | Reserved |

**Bit 23 – SEXTTOEN** Client SCL Low Extend Time-Out

This bit enables the client SCL low extend time-out. If SCL is cumulatively held low for greater than 25ms from the initial START to a STOP, the host will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.
SB or MB will be set as normal, but CLKHOLD will be release. The MEXTTOUT and BUSERR status bits will be set.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Time-out disabled |
| 1 | Time-out enabled |

**Bit 22 – MEXTTOEN** Host SCL Low Extend Time-Out

This bit enables the host SCL low extend time-out. If SCL is cumulatively held low for greater than 10ms from START-to-ACK, ACK-to-ACK, or ACK-to-STOP the host will release its clock hold if enabled, and complete the current transaction. A STOP will automatically be transmitted.
SB or MB will be set as normal, but CLKHOLD will be released. The MEXTTOUT and BUSERR status bits will be set.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Time-out disabled |
| 1 | Time-out enabled |

**Bits 21:20 – SDAHOLD[1:0]** SDA Hold Time

These bits define the SDA hold time with respect to the negative edge of SCL.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIS | Disabled |
| 0x1 | 75NS | 50-100ns hold time |
| 0x2 | 450NS | 300-600ns hold time |
| 0x3 | 600NS | 400-800ns hold time |

**Bit 16 – PINOUT** Pin Usage

This bit set the pin usage to either two- or four-wire operation:
This bit is not synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | 4-wire operation disabled. |
| 1 | 4-wire operation enabled. |

**Bit 7 – RUNSTDBY** Run in Standby

This bit defines the functionality in standby sleep mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | GCLK_SERCOMx_CORE is disabled and the $I^2C$ host will not operate in standby sleep mode. |
| 1 | GCLK_SERCOMx_CORE is enabled in all sleep modes. |

**Bits 4:2 – MODE[2:0]** Operating Mode

These bits must be written to 0x5 to select the $I^2C$ host serial communication interface of the SERCOM.
These bits are not synchronized.

**Bit 1 – ENABLE** Enable

Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the Synchronization Enable Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled or being disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset

Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the SERCOM, except DBGCTRL, to their initial state, and the SERCOM will be disabled.
Writing '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded. Any register write access during the ongoing reset will result in an APB error. Reading any register will return the reset value of the register.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ACKACT | CMD[1:0] | |
| Access | | | | | | R/W | W | W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | QCEN | SMEN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bit 18 – ACKACT** Acknowledge Action

This bit defines the $I^2C$ host's acknowledge behavior after a data byte is received from the $I^2C$ client. The acknowledge action is executed when a command is written to CTRLB.CMD, or if Smart mode is enabled (CTRLB.SMEN is written to one), when DATA.DATA is read.
This bit is not enable-protected.
This bit is not write-synchronized.

| Value | Description |
|---|---|
| 0 | Send ACK. |
| 1 | Send NACK. |

**Bits 17:16 – CMD[1:0]** Command

Writing these bits triggers a host operation as described below. The CMD bits are strobe bits, and always read as zero. The acknowledge action is only valid in Host Read mode. In Host Write mode, a command will only result in a repeated Start or Stop condition. The CTRLB.ACKACT bit and the CMD bits can be written at the same time, and then the acknowledge action will be updated before the command is triggered.
Commands can only be issued when either the Client on Bus Interrupt flag (INTFLAG.SB) or Host on Bus Interrupt flag (INTFLAG.MB) is '1'.
If CMD 0x1 is issued, a repeated start will be issued followed by the transmission of the current address in ADDR.ADDR. If another address is desired, ADDR.ADDR must be written instead of the CMD bits. This will trigger a repeated start followed by transmission of the new address.
Issuing a command will set the System Operation bit in the Synchronization Busy register (SYNCBUSY.SYSOP).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 36-4.** Command Description

| CMD[1:0] | Direction | Action |
|---|---|---|
| 0x0 | X | (No action) |
| 0x1 | X | Execute acknowledge action succeeded by repeated Start |
| 0x2 | 0 (Write) | No operation |
| | 1 (Read) | Execute acknowledge action succeeded by a byte read operation |
| 0x3 | X | Execute acknowledge action succeeded by issuing a Stop condition |

These bits are not enable-protected.

**Bit 9 – QCEN** Quick Command Enable
This bit is not write-synchronized.

| Value | Description |
|---|---|
| 0 | Quick Command is disabled. |
| 1 | Quick Command is enabled. |

**Bit 8 – SMEN** Smart Mode Enable
When Smart mode is enabled, acknowledge action is sent when DATA.DATA is read.
This bit is not write-synchronized.

| Value | Description |
|---|---|
| 0 | Smart mode is disabled. |
| 1 | Smart mode is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 36.10.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DATA32B |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bit 24 – DATA32B** Data 32 Bit
This bit enables 32-bit data writes and reads to/from the DATA register.

| Value | Description |
|---|---|
| 0 | Data transactions to/from DATA are 8-bit in size |
| 1 | Data transactions to/from DATA are 32-bit in size |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.4. Baud Rate

**Name:** BAUD
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | HSBAUDLOW[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | HSBAUD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BAUDLOW[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:24 – HSBAUDLOW[7:0]** High Speed Host Baud Rate Low

HSBAUDLOW non-zero: HSBAUDLOW indicates the SCL low time in High-speed mode according to $\mathrm{HSBAUDLOW} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{LOW}} - 1$

HSBAUDLOW equal to zero: The HSBAUD register is used to time $T_{LOW}$, $T_{HIGH}$, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$. $T_{BUF}$ is timed by the BAUD register.

**Bits 23:16 – HSBAUD[7:0]** High Speed Host Baud Rate

This bit field indicates the SCL high time in High-speed mode according to the following formula. When HSBAUDLOW is zero, $T_{LOW}$, $T_{HIGH}$, $T_{SU;STO}$, $T_{HD;STA}$ and $T_{SU;STA}$ are derived using this formula. $T_{BUF}$ is timed by the BAUD register.

$\mathrm{HSBAUD} = f_{\mathrm{GCLK}} \cdot T_{\mathrm{HIGH}} - 1$

**Bits 15:8 – BAUDLOW[7:0]** Host Baud Rate Low

If this bit field is non-zero, the SCL low time will be described by the value written.

For more information on how to calculate the frequency, see SERCOM Clock Generation – Baud-Rate Generator.

**Bits 7:0 – BAUD[7:0]** Host Baud Rate

This bit field is used to derive the SCL high time if BAUD.BAUDLOW is non-zero. If BAUD.BAUDLOW is zero, BAUD will be used to generate both high and low periods of the SCL.

For more information on how to calculate the frequency, see SERCOM Clock Generation – Baud-Rate Generator.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | | | SB | MB |
| Access | R/W | | | | | | R/W | R/W |
| Reset | 0 | | | | | | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 1 – SB** Client on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Client on Bus Interrupt Enable bit, which disables the Client on Bus interrupt.

| Value | Description |
|---|---|
| 0 | The Client on Bus interrupt is disabled. |
| 1 | The Client on Bus interrupt is enabled. |

**Bit 0 – MB** Host on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Host on Bus Interrupt Enable bit, which disables the Host on Bus interrupt.

| Value | Description |
|---|---|
| 0 | The Host on Bus interrupt is disabled. |
| 1 | The Host on Bus interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x16
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | | | SB | MB |
| Access | R/W | | | | | | R/W | R/W |
| Reset | 0 | | | | | | 0 | 0 |

**Bit 7 – ERROR** Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | Error interrupt is disabled. |
| 1 | Error interrupt is enabled. |

**Bit 1 – SB** Client on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Client on Bus Interrupt Enable bit, which enables the Client on Bus interrupt.

| Value | Description |
|---|---|
| 0 | The Client on Bus interrupt is disabled. |
| 1 | The Client on Bus interrupt is enabled. |

**Bit 0 – MB** Host on Bus Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the Host on Bus Interrupt Enable bit, which enables the Host on Bus interrupt.

| Value | Description |
|---|---|
| 0 | The Host on Bus interrupt is disabled. |
| 1 | The Host on Bus interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | | | | SB | MB |
| Access | R/W | | | | | | R/W | R/W |
| Reset | 0 | | | | | | 0 | 0 |

**Bit 7 – ERROR** Error

This flag is cleared by writing '1' to it.
This bit is set when any error is detected. Errors that will set this flag have corresponding status bits in the STATUS register. These status bits are LENERR, SEXTTOUT, MEXTTOUT, LOWTOUT, ARBLOST, and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 1 – SB** Client on Bus

The Client on Bus flag (SB) is set when a byte is successfully received in Host Read mode, for example, no arbitration lost or bus error occurred during the operation. When this flag is set, the host forces the SCL line low, stretching the $I^2C$ clock period. The SCL line will be released and SB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the SB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.

**Bit 0 – MB** Host on Bus

This flag is set when a byte is transmitted in Host Write mode. The flag is set regardless of the occurrence of a bus error or an Arbitration Lost condition. MB is also set when arbitration is lost during sending of NACK in Host Read mode, or when issuing a Start condition if the bus state is unknown. When this flag is set and arbitration is not lost, the host forces the SCL line low, stretching the $I^2C$ clock period. The SCL line will be released and MB will be cleared on one of the following actions:

- Writing to ADDR.ADDR
- Writing to DATA.DATA
- Reading DATA.DATA when Smart mode is enabled (CTRLB.SMEN)
- Writing a valid command to CTRLB.CMD

Writing '1' to this bit location will clear the MB flag. The transaction will not continue or be terminated until one of the above actions is performed.
Writing '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.8. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | LENERR | SEXTTOUT | MEXTTOUT |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CLKHOLD | LOWTOUT | BUSSTATE[1:0] | | | RXNACK | ARBLOST | BUSERR |
| Access | R | R/W | R/W | R/W | | R | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bit 10 – LENERR** Transaction Length Error
This bit is set when automatic length is used for a DMA and/or 32-bit transaction and the client sends a NACK before ADDR.LEN bytes have been written by the host.
Writing '1' to this bit location will clear STATUS.LENERR. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

**Bit 9 – SEXTTOUT** Client SCL Low Extend Time-Out
This bit is set if a client SCL low extend time-out occurs.
This bit is automatically cleared when writing to the ADDR register.
Writing '1' to this bit location will clear SEXTTOUT. Normal use of the $I^2C$ interface does not require the SEXTTOUT flag to be cleared by this method.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

**Bit 8 – MEXTTOUT** Host SCL Low Extend Time-Out
This bit is set if a Host SCL low time-out occurs.
Writing '1' to this bit location will clear STATUS.MEXTTOUT. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.

**Bit 7 – CLKHOLD** Clock Hold
This bit is set when the host is holding the SCL line low, stretching the $I^2C$ clock. Software should consider this bit when INTFLAG.SB or INTFLAG.MB is set.
This bit is cleared when the corresponding Interrupt flag is cleared and the next operation is given.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

**Bit 6 – LOWTOUT** SCL Low Time-Out
This bit is set if an SCL low time-out occurs.
Writing '1' to this bit location will clear this bit. This flag is automatically cleared when writing to the ADDR register.
Writing '0' to this bit has no effect.
This bit is not write-synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 5:4 – BUSSTATE[1:0]** Bus State

These bits indicate the current I²C Bus state.
When in UNKNOWN state, writing 0x1 to BUSSTATE forces the bus state into the IDLE state. The bus state cannot be forced into any other state.
Writing BUSSTATE to idle will set SYNCBUSY.SYSOP.

| Value | Name | Description |
|---|---|---|
| 0x0 | UNKNOWN | The Bus state is unknown to the I²C host and will wait for a Stop condition to be detected or wait to be forced into an Idle state by software |
| 0x1 | IDLE | The Bus state is waiting for a transaction to be initialized |
| 0x2 | OWNER | The I²C host is the current owner of the bus |
| 0x3 | BUSY | Some other I²C host owns the bus |

**Bit 2 – RXNACK** Received Not Acknowledge

This bit indicates whether the last address or data packet sent was acknowledged or not.
Writing '0' to this bit has no effect.
Writing '1' to this bit has no effect.
This bit is not write-synchronized.

| Value | Description |
|---|---|
| 0 | Client responded with ACK. |
| 1 | Client responded with NACK. |

**Bit 1 – ARBLOST** Arbitration Lost

This bit is set if arbitration is lost while transmitting a high data bit or a NACK bit, or while issuing a Start or Repeated Start condition on the bus. The Host on Bus Interrupt flag (INTFLAG.MB) will be set when STATUS.ARBLOST is set.
Writing the ADDR.ADDR register will automatically clear STATUS.ARBLOST.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.
This bit is not write-synchronized.

**Bit 0 – BUSERR** Bus Error

This bit indicates that an illegal Bus condition has occurred on the bus, regardless of bus ownership. An illegal Bus condition is detected if a protocol violating start, repeated start or stop is detected on the I²C bus lines. A Start condition directly followed by a Stop condition is one example of a protocol violation. If a time-out occurs during a frame, this is also considered a protocol violation, and will set BUSERR.
If the I²C host is the bus owner at the time a bus error occurs, STATUS.ARBLOST and INTFLAG.MB will be set in addition to BUSERR.
Writing the ADDR.ADDR register will automatically clear the BUSERR flag.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear it.
This bit is not write-synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.9. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SYSOP | ENABLE | SWRST |
| Access | | | | | | R | R | R |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – SYSOP** System Operation Synchronization Busy
Writing CTRLB.CMD, STATUS.BUSSTATE, ADDR, or DATA when the SERCOM is enabled requires synchronization. When written, the SYNCBUSY.SYSOP bit will be set until synchronization is complete.

| Value | Description |
|---|---|
| 0 | System operation synchronization is not busy. |
| 1 | System operation synchronization is busy. |

**Bit 1 – ENABLE** SERCOM Enable Synchronization Busy
Enabling and disabling the SERCOM (CTRLA.ENABLE) requires synchronization. When written, the SYNCBUSY.ENABLE bit will be set until synchronization is complete.

| Value | Description |
|---|---|
| 0 | Enable synchronization is not busy. |
| 1 | Enable synchronization is busy. |

**Bit 0 – SWRST** Software Reset Synchronization Busy
Resetting the SERCOM (CTRLA.SWRST) requires synchronization. When written, the SYNCBUSY.SWRST bit will be set until synchronization is complete.

| Value | Description |
|---|---|
| 0 | SWRST synchronization is not busy. |
| 1 | SWRST synchronization is busy. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.10. Address

**Name:** ADDR
**Offset:** 0x24
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | LEN[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TENBITEN | HS | LENEN | | | ADDR[10:8] | | |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – LEN[7:0]** Transaction Length
These bits define the transaction length of a DMA and/or 32-bit transaction from 0 to 255 bytes. The Transfer Length Enable (LENEN) bit must be written to '1' in order to use DMA.

**Bit 15 – TENBITEN** Ten Bit Addressing Enable
This bit enables 10-bit addressing. This bit can be written simultaneously with ADDR to indicate a 10-bit or 7-bit address transmission.

| Value | Description |
|---|---|
| 0 | 10-bit addressing disabled. |
| 1 | 10-bit addressing enabled. |

**Bit 14 – HS** High Speed
This bit enables High-speed mode for the current transfer from repeated START to STOP. This bit can be written simultaneously with ADDR for a high speed transfer.

| Value | Description |
|---|---|
| 0 | High-speed transfer disabled. |
| 1 | High-speed transfer enabled. |

**Bit 13 – LENEN** Transfer Length Enable

| Value | Description |
|---|---|
| 0 | Automatic transfer length disabled. |
| 1 | Automatic transfer length enabled. |

**Bits 10:0 – ADDR[10:0]** Address
When ADDR is written, the consecutive operation will depend on the bus state:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

UNKNOWN: INTFLAG.MB and STATUS.BUSERR are set, and the operation is terminated.
BUSY: The $I^2C$ host will await further operation until the bus becomes IDLE.
IDLE: The $I^2C$ host will issue a start condition followed by the address written in ADDR. If the address is acknowledged, SCL is forced and held low, and STATUS.CLKHOLD and INTFLAG.MB are set.
OWNER: A repeated start sequence will be performed. If the previous transaction was a read, the acknowledge action is sent before the repeated start bus condition is issued on the bus. Writing ADDR to issue a repeated start is performed while INTFLAG.MB or INTFLAG.SB is set. STATUS.BUSERR, STATUS.ARBLOST, INTFLAG.MB and INTFLAG.SB will be cleared when ADDR is written.
The ADDR register can be read at any time without interfering with ongoing bus activity, as a read access does not trigger the host logic to perform any bus protocol related operations.
The $I^2C$ host control logic uses bit 0 of ADDR as the bus protocol's read/write flag (R/W); 0 for write and 1 for read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.11. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Data

The Host data register I/O location (DATA) provides access to the Host transmit and receive data buffers. Reading valid data or writing data to be transmitted can be successfully done only when SCL is held low by the Host (STATUS.CLKHOLD is set). An exception is reading the last data byte after the stop condition has been sent.

Accessing DATA.DATA auto-triggers $I^2C$ bus operations. The operation performed depends on the state of CTRLB.ACKACT, CTRLB.SMEN and the type of access (read/write).

When CTRLC.DATA32B=1, read and write transactions from/to the DATA register are 32 bit in size. Otherwise, reads and writes are 8 bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.10.12. Debug Control

**Name:** DBGCTRL
**Offset:** 0x30
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGSTOP |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGSTOP** Debug Stop Mode
This bit controls functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The baud-rate generator continues normal operation when the CPU is halted by an external debugger. |
| 1 | The baud-rate generator is halted when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 37. QSPI - Quad Serial Peripheral Interface

## 37.1. Overview

The Quad SPI Interface (QSPI) circuit is a synchronous serial data link that provides communication with external devices in Host mode.

The QSPI can be used in “SPI mode” to interface serial peripherals, such as ADCs, DACs, LCD controllers and sensors, or in “Serial Memory Mode” to interface serial Flash memories.

The QSPI allows the system to execute code directly from a serial Flash memory (XIP) without code shadowing to SRAM. The serial Flash memory mapping is seen in the system as other memories (ROM, SRAM, DRAM, embedded Flash memories, etc.,).

With the support of the quad-SPI protocol, the QSPI allows the system to use high performance serial Flash memories which are small and inexpensive, in place of larger and more expensive parallel Flash memories.

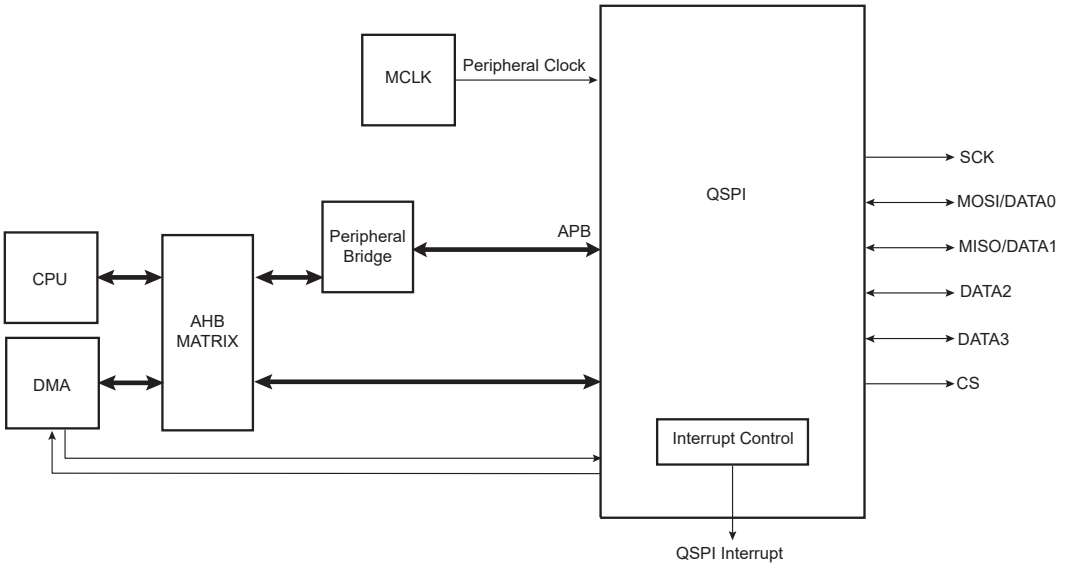
## 37.2. Features

- Host SPI Interface:
  - Programmable Clock Phase and Clock Polarity
  - Programmable transfer delays between consecutive transfers, between clock and data, between deactivation and activation of chip select (CS)
- SPI Mode:
  - To use serial peripherals, such as ADCs, DACs, LCD controllers, CAN controllers, and sensors
  - 8-bit, 16-bit, or 32-bit programmable data length
- Serial Memory Mode:
  - To use serial Flash memories operating in single-bit SPI, Dual SPI and Quad SPI
  - Supports “execute in place” (XIP). The system can execute code directly from a Serial Flash memory.
  - Flexible Instruction register, to be compatible with all Serial Flash memories
  - 32-bit Address mode (default is 24-bit address) to support Serial Flash memories larger than 128 Mbit
  - Continuous Read mode
  - Scrambling/Unscrambling “On-the-Fly”
  - Double data rate support (read only)
- Connection to DMA Channel Capabilities Optimizes Data Transfers
  - One channel for the receiver and one channel for the transmitter
- Register Write Protection


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 37.3. Block Diagram

**Figure 37-1.** QSPI Block Diagram




## 37.4. Signal Description

**Table 37-1.** Quad-SPI Signals

| Signal | Description | Type |
|---|---|---|
| SCK | Serial Clock | Output |
| CS | Chip Select | Output |
| MOSI(DATA0) | Data Output (Data Input Output 0) | Output (Input/Output) |
| MISO(DATA1) | Data Input (Data Input Output 1) | Input (Input/Output) |
| DATA2 | Data Input Output 2 | Input/Output |
| DATA3 | Data Input Output 3 | Input/Output |

**Note:** MOSI and MISO are used for single-bit SPI operation

**Note:** DATA0-DATA1 are used for Dual SPI operation

**Note:** DATA0-DATA3 are used for Quad SPI operation

Refer to the pinout table for details on the pin mapping for this peripheral. One signal can be mapped to one of several pins.

## 37.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 37.5.1. I/O Lines

Using the QSPI I/O lines requires the I/O pins to be configured.

**Related Links**

PORT - I/O Pin Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.5.2. Power Management

The QSPI will continue to operate in any Sleep mode where the selected source clock is running. The QSPI interrupts can be used to wake up the device from sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

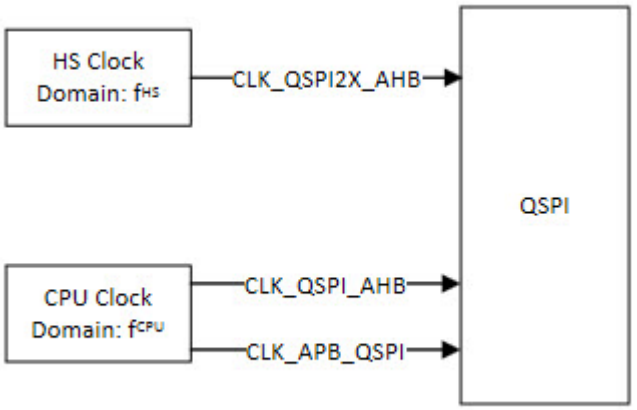
### 37.5.3. Clocks

The QSPI bus clock (CLK_QSPI_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_QSPI_APB can be found in the Peripheral Clock Masking section in the MCLK chapter.

An AHB clock (CLK_QSPI_AHB) is required to clock the QSPI. This clock can be enabled and disabled in the Main Clock module, and the default state of CLK_QSPI_AHB can be found in the Peripheral Clock Masking section in the MCLK chapter.

A FAST clock (CLK_QSPI2X_AHB) is required to clock the QSPI. This clock can be enabled and disabled in the Main Clock module, and the default state of CLK_QSPI2X_AHB can be found in the Peripheral Clock Masking section in the MCLK chapter. This clock is derived from the High-Speed Clock Domain (HS Clock Domain, frequency $f_{HS}$).

**Figure 37-2.** QSPI Clock Organization




**Important:** The CLK_QSPI2x_AHB must be 2 times faster to CLK_QSPI_AHB when the QSPI is operated in DDR mode. In SDR, the CLK_QSPI2x_AHB is not used.

CLK_QSPI_APB, CLK_QSPI_AHB, and CLK_QSPI2X_AHB, respectively, are all synchronous, but can be divided by a prescaler and may run even when the module clock is turned off.

**Related Links**

MCLK – Main Clock
Peripheral Clock Masking

### 37.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). Using the QSPI DMA requests requires the DMA Controller to be configured first.

**Note:** DMAC write access must be 32-bit aligned. If a single byte is to be written in a 32-bit word, the rest of the word must be filled with 'ones'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
DMAC – Direct Memory Access Controller

### 37.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the QSPI interrupts requires the interrupt controller to be configured first. Refer to the Nested Vector Interrupt Controller section for details.

**Related Links**
Nested Vector Interrupt Controller

### 37.5.6. Events

Not applicable.

### 37.5.7. Debug Operation

When the CPU is halted in debug mode the QSPI continues normal operation. If the QSPI is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

### 37.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following registers:

- Control A (CTRLA) register
- Transmit Data (TXDATA) register
- Interrupt Flag Status and Clear (INTFLAG) register
- Interrupt Flag Status and Clear (INTFLAG) register

PAC write-protection is denoted by the `'PAC Write-Protection' property in the register description.

Write-protection does not apply to accesses through an external debugger.

## 37.6. Functional Description

### 37.6.1. Principle of Operation

The QSPI is a high-speed synchronous data transfer interface. It allows high-speed communication between the device and peripheral or serial memory devices.

The QSPI operates as a Host. It initiates and controls all data transactions.

When transmitting, the TXDATA register can be loaded with the next character to be transmitted during the current transmission.

When receiving, the data is transferred to the RXDATA register, and the receiver is ready for a new character.

### 37.6.2. Basic Operation

#### 37.6.2.1. Initialization

After Power-On Reset, this peripheral is enabled .

#### 37.6.2.2. Enabling, Disabling, and Resetting

The peripheral is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE).

The peripheral is disabled by writing a '0' to CTRLA.ENABLE.

The peripheral is reset by writing a '1' to the Software Reset bit (CTRLA.SWRST).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.6.3. Transfer Data Rate

By default, the QSPI module is enabled in Single Data Rate (SDR) mode. In this operating mode, the CLK_QSPI2X_AHB clock is not used and can be disabled.

The Dual Data Rate (DDR) operating mode (read only) is enabled by writing a '1' to the Double Data Rate Enable bit in the Instruction Frame register (INSTRFRAME.DDREN). This operating mode requires the CLK_QSPI2X_AHB clock and must be enabled before writing the DDREN bit.

### 37.6.4. Serial Clock Baudrate

The QSPI Baud rate clock is generated by dividing the module clock (CLK_QSPI_AHB) by a value between 1 and 255.

This allows a maximum operating baud rate at up to Host Clock and a minimum operating baud rate of CLK_QSPI_AHB divided by 256.

### 37.6.5. Serial Clock Phase and Polarity

Four combinations of polarity and phase are available for data transfers. Writing the Clock Polarity bit in the QSPI Baud register (BAUD.CPOL) selects the polarity. The Clock Phase bit in the BAUD register programs the clock phase (BAUD.CPHA). These two parameters determine the edges of the clock signal on which data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations.
**Note:** The polarity/phase combinations are incompatible. Therefore, the interfaced Client must use the same parameter values to communicate.

All combinations of polarity and phase are available when QSPI operates in SPI mode (CTRLB.MODE = 0).

In QSPI Serial Memory mode (CTRLB.MODE = 1), only Mode 0 is supported.

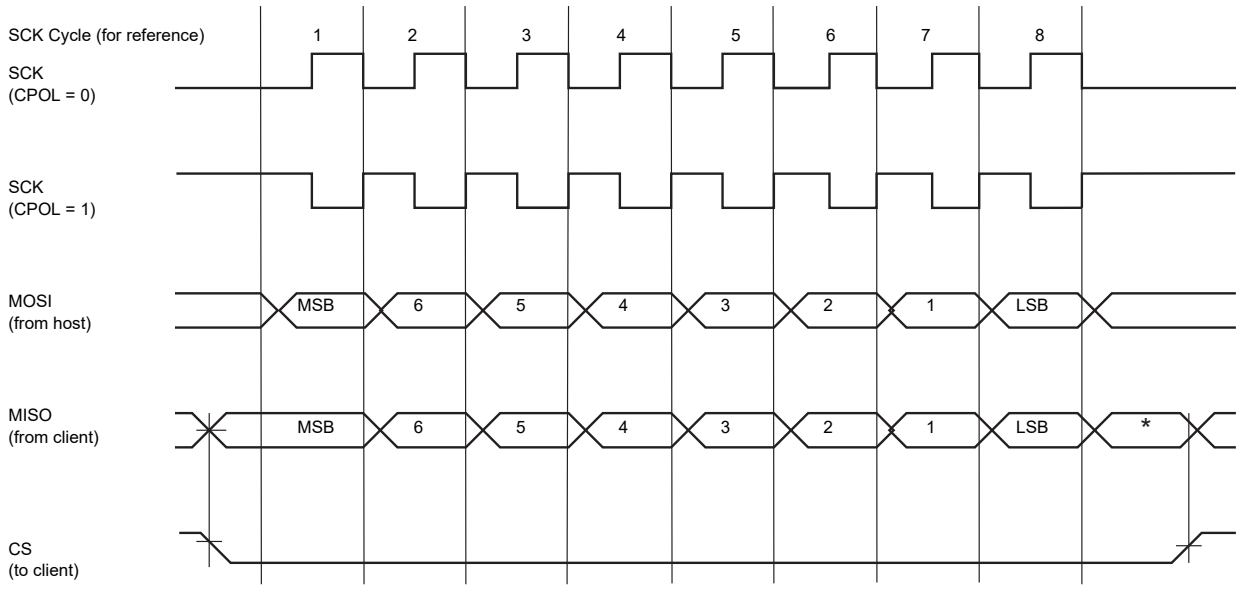
in QSPI Dual Data Rate transfer mode (INSTRFRAME.DDREN = 1), only Mode 0 is supported.

**Table 37-2.** QSPI Transfer Mode

| Mode | BAUD.CPOL | BAUD.CPHA | Shift SCK Edge | Capture SCK Edge | SCK Inactive Level |
|---|---|---|---|---|---|
| 0 | 0 | 0 | Falling | Falling | Low |
| 1 | 0 | 1 | Rising | Rising | Low |
| 2 | 1 | 0 | Rising | Rising | High |
| 3 | 1 | 1 | Falling | Falling | High |

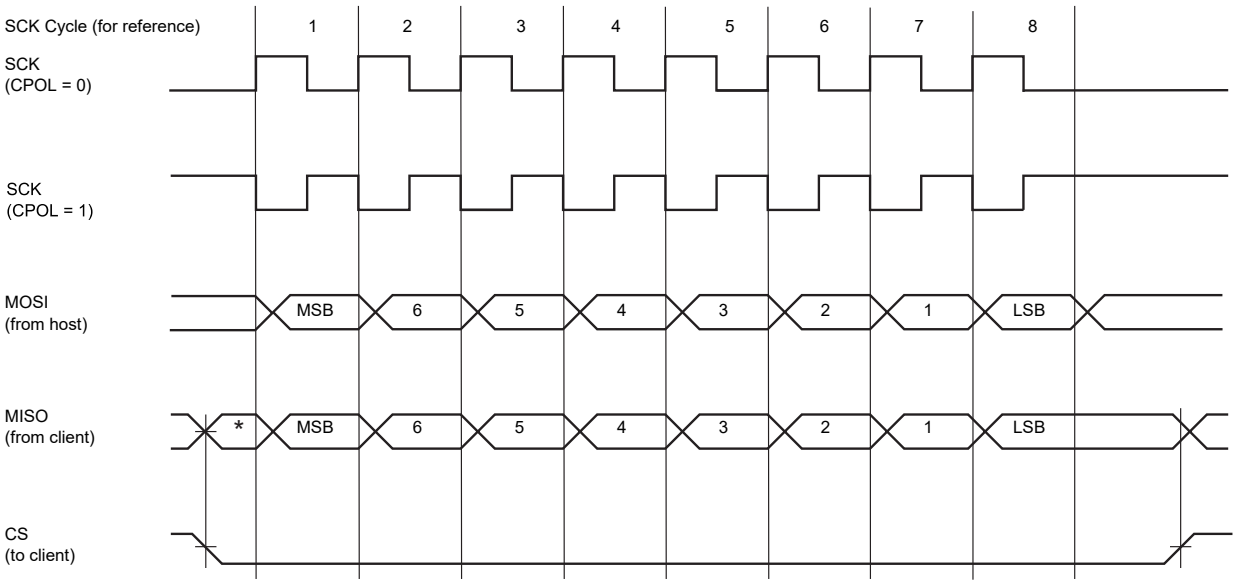

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 37-3.** QSPI Transfer Modes (BAUD.CPHA = 0, 8-bit transfer)




* Not defined, but normally MSB of previous character received

**Figure 37-4.** QSPI Transfer Modes (BAUD.CPHA = 1, 8-bit transfer)




* Not defined, but normally LSB of previous character received

### 37.6.6. Transfer Delays

The QSPI supports several consecutive transfers while the chip select is active. Three delays can be programmed to modify the transfer waveforms:

- The delay between the inactivation and the activation of CS is programmed by writing the Minimum Inactive CS Delay bit field in the Control B register (CTRLB.DLYCS), allowing to tune the minimum time of CS at high level.
- The delay between consecutive transfers is programmed by writing the Delay Between Consecutive Transfers bit field in the Control B register (CTRLB.DLYBCT), allowing to insert a delay

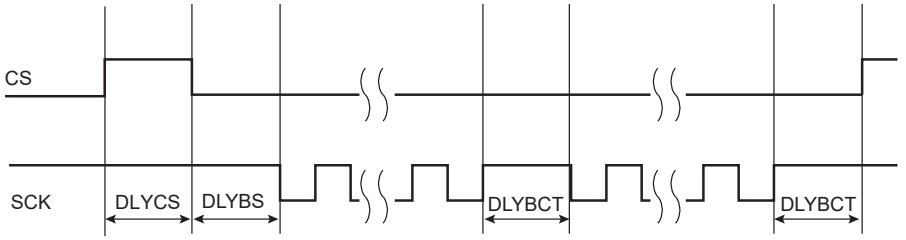

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

between two consecutive transfers. In Serial Memory mode, this delay is not programmable and DLYBCT settings are ignored.

- The delay before SCK is programmed by writing the Delay Before SCK bit field in the BAUD register (BAUD.DLYBS), allowing to delay the start of SPCK after the chip select has been asserted.

These delays allow the QSPI to be adapted to the interfaced peripherals and their speed and bus release time.

**Figure 37-5.** Programmable Delay




### 37.6.7. QSPI SPI Mode

In this mode, the QSPI acts as a regular SPI Host.

To activate this mode, the MODE bit in Control B register must be cleared (CTRLB.MODE=0).

#### 37.6.7.1. SPI Mode Operations

The QSPI in standard SPI mode operates on the clock generated by the internal programmable baud rate generator. It fully controls the data transfers to and from the Client connected to the SPI bus. The QSPI drives the chip select line to the Client (CS) and the serial clock signal (SCK).

The QSPI features a single internal shift register and two holding registers: the Transmit Data Register (TXDATA) and the Receive Data Register (RXDATA). The holding registers maintain the data flow at a constant rate.

After enabling the QSPI, a data transfer begins when the processor writes to the TXDATA. The written data is immediately transferred into the internal shift register and transfer on the SPI bus starts. While the data in the internal shift register is shifted on the MOSI line, the MISO line is sampled and shifted into the internal shift register. Receiving data cannot occur without transmitting data.

If new data is written in TXDATA during the transfer, it stays in TXDATA until the current transfer is completed. Then, the received data is transferred from the internal shift register to the RXDATA, the data in TXDATA is loaded into the internal shift register, and a new transfer starts.

The transfer of data written in TXDATA in the internal shift register is indicated by the Transmit Data Register Empty (DRE) bit in the Interrupt Flag Status and Clear register (INTFLAG.DRE). When new data is written in TXDATA, this bit is cleared. The DRE bit is used to trigger the Transmit DMA channel.

The end of transfer is indicated by the Transmission Complete flag (INTFLAG.TXC). If the transfer delay for the last transfer was configured to be greater than 0 (CTRLB.DLYBCT), TXC is set after the completion of the delay. The module clock (CLK_QSPI_AHB) can be switched off at this time.

Ongoing transfer of received data from the internal shift register into RXDATA is indicated by the Receive Data Register Full flag (INTFLAG.RXC). When the received data is read, the RXC bit is cleared.

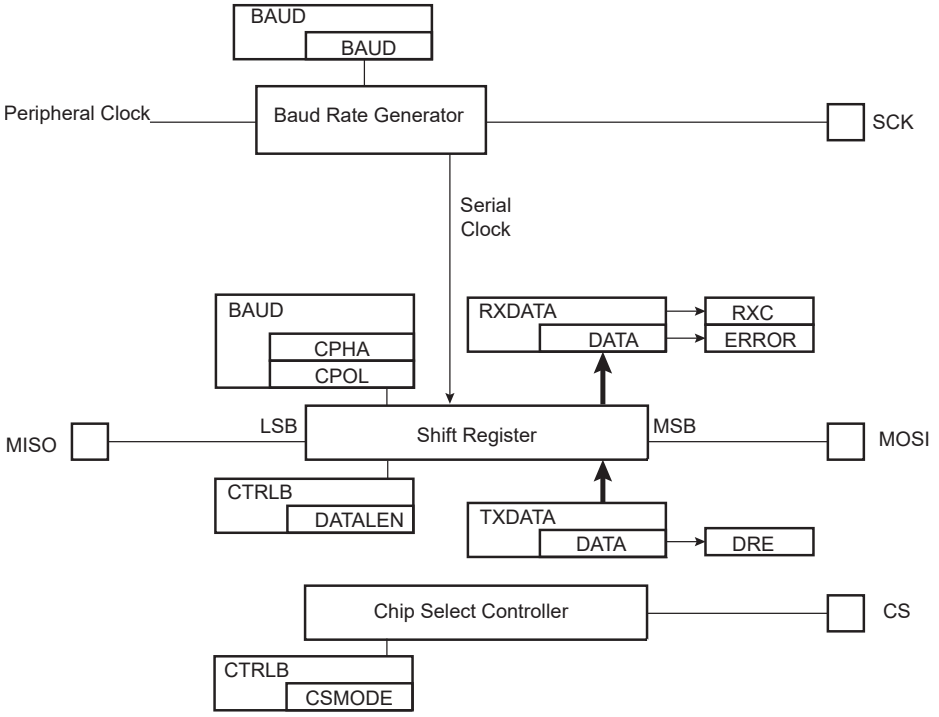
If the RXDATA has not been read before new data is received, the Overrun Error flag in INTFLAG register (INTFLAG.ERROR) is set. As long as this flag is set, data is loaded in RXDATA.

The SPI Mode Block Diagram shows a flow chart describing how transfers are handled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.6.7.2. SPI Mode Block Diagram

**Figure 37-6.** SPI Mode Block Diagram

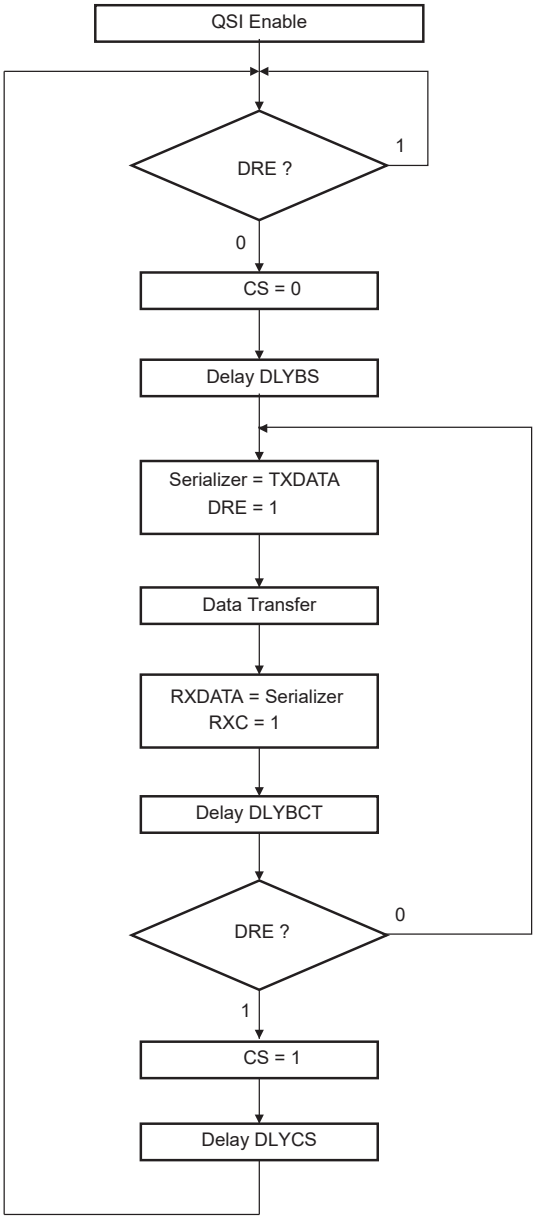




© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.6.7.3. SPI Mode Flow Diagram

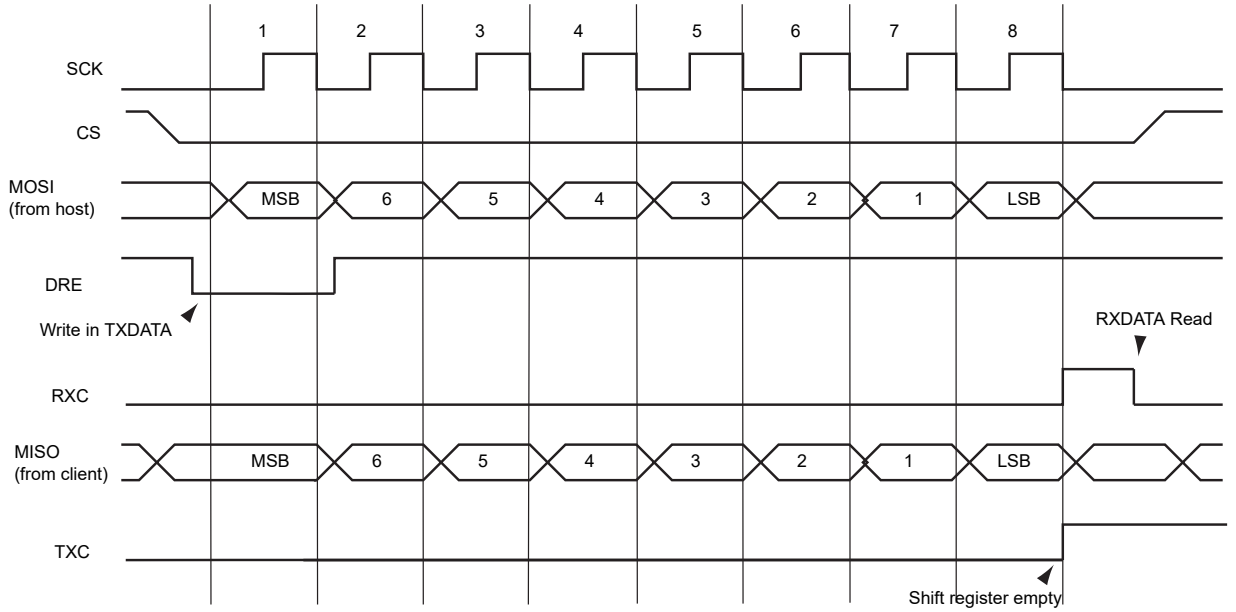
**Figure 37-7.** SPI Mode Flow Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 37-8.** Interrupt Flags Behaviour




### 37.6.7.4. Peripheral Deselection with DMA

When the Direct Memory Access Controller is used, the Chip Select line will remain low during the whole transfer since the Transmit Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) is managed by the DMA itself. The reloading of the TXDATA by the DMA is done as soon as INTFLAG.DRE flag is set. In this case, setting the Chip Select Mode bit field in the Control B register (CTRLB.CSMODE) to 0x1 is not mandatory.

However, it may happen that when other DMA channels connected to other peripherals are in use as well, the QSPI DMA could be delayed by another DMA transfer with a higher priority on the bus. Having DMA buffers in slower memories like flash memory or SDRAM (compared to fast internal SRAM), may lengthen the reload time of the TXDATA by the DMA as well. This means that TXDATA might not be reloaded in time to keep the Chip Select line low. In this case the Chip Select line may toggle between data transfer and according to some SPI Client devices, and the communication might get lost. Writing CTRLB.CSMODE=0x1 can prevent this loss.

When CTRLB.CSMODE=0x0, the CS does not rise in all cases between two transfers on the same peripheral. During a transfer on a Chip Select, the INTFLAG.DRE flag is raised as soon as the content of the TXDATA is transferred into the internal shifter. When this flag is detected the TXDATA can be reloaded. if this reload occurs before the end of the current transfer and if the next transfer is performed on the same Chip Select as the current transfer, the Chip Select is not de-asserted between the two transfers. This may lead to difficulties for interfacing with some serial peripherals requiring the Chip Select to be de-asserted after each transfer. To facilitate interfacing with such devices, it is recommended to write CTRLB.CSMODE to 0x2.

### 37.6.7.5. Peripheral Deselection without DMA

During multiple data transfers on a Chip Select without the DMA, the TXDATA is loaded by the processor, and the Transmit Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) rises as soon as the content of the RXDATA is transferred into the internal shift register. When this flag is detected high, the TXDATA can be reloaded. If this reload-by-processor occurs before the end of the current transfer and if the next transfer is performed on the same Chip Select as the current transfer, the Chip Select is not de-asserted between the two transfers.

Depending on the application software handling the flags or servicing other interrupts or other tasks, the processor may not reload the TXDATA in time to keep the Chip Select active (low). A null


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Delay Between Consecutive Transfer bit field value in the CTRLB register (CTRLB.DLYBCT) will give even less time for the processor to reload the TXDATA. With some SPI Client peripherals, requiring the Chip Select line to remain active (low) during a full set of transfers might lead to communication errors.

To facilitate interfacing with such devices, the Chip Select Mode bit field in the CTRLB register (CTRLB.CSMODE) can be written to 0x1. This allows the Chip Select lines to remain in their current state (low = active) until the end of transfer is indicated by the Last Transfer bit in the CTRLA register (CTRLA.LASTXFER). Even if the TXDATA is not reloaded the Chip Select will remain active. To have the Chip Select line rise at the end of the last data transfer, the LASTXFER bit in the CTRLA must be set before writing the last data to transmit into the TXDATA.

### 37.6.8. QSPI Serial Memory Mode

In this mode the QSPI acts as a serial Flash memory controller. The QSPI can be used to read data from the serial Flash memory allowing the CPU to execute code from it (XIP execute in place). The QSPI can also be used to control the serial Flash memory (Program, Erase, Lock, and so on) by sending specific commands. In this mode, the QSPI is compatible with single-bit SPI, Dual-SPI and Quad-SPI protocols.

To activate this mode, the MODE bit in Control B register must be set to one (CTRLB.MODE = 1).

In serial memory mode, data cannot be transferred by the TXDATA and the RXDATA, but by writing or reading the QSPI memory space (0x0400 0000 – 0x0500 0000).

**Important:** QSPI memory space region can be cached to improve data transfer speed.
However, external Flash devices which have command/status registers mapped in the QSPI memory space region must be managed carefully by applying any one of the following configurations:

- Data cache must be disabled.
- If data cache is required, then cache line must be invalidated before reading the status register.

#### 37.6.8.1. Instruction Frame

In order to control serial Flash memories, the QSPI is able to sent instructions by the SPI bus (ex: READ, PROGRAM, ERASE, LOCK, etc.). Because instruction set implemented in serial Flash memories is memory vendor dependent, the QSPI includes a complete instruction registers, which makes it very flexible and compatible with all serial Flash memories.

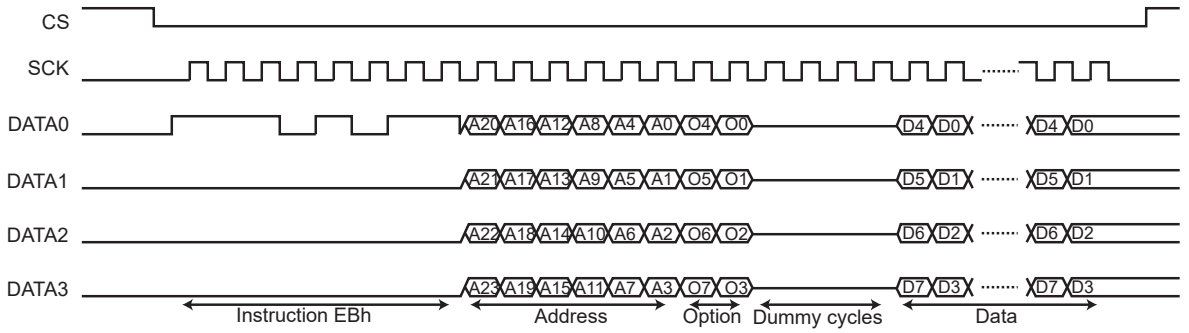
An instruction frame includes:

- **An instruction code (size: 8 bits):** The instruction can be optional in some cases
- **An address (size: 24 bits or 32 bits):** The address is optional but is required by instructions such as READ, PROGRAM, ERASE, LOCK. By default the address is 24 bits long, but it can be 32 bits long to support serial Flash memories larger than 128 Mbit (16 Mbyte).
- **An option code (size: 1/2/4/8 bits):** The option code is optional but is useful for activate the “XIP mode” or the “Continuous Read Mode” for READ instructions, in some serial Flash memory devices. These modes allow to improve the data read latency.
- **Dummy cycles:** Dummy cycles are optional but required by some READ instructions
- **Data bytes are optional:** Data bytes are present for data transfer instructions such as READ or PROGRAM

The instruction code, the address/option and the data can be sent with Single-bit SPI, Dual SPI or Quad SPI protocols.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 37-9.** Instruction Frame




### 37.6.8.2. Instruction Frame Sending

To send an instruction frame, the user must first configure the address to send by writing the field ADDR in the Instruction Address Register (INSTRADDR.ADDR). This step is required if the instruction frame includes an address and no data. When data is present, the address of the instruction is defined by the address of the data accesses in the QSPI memory space, and not by the INSTRADDR register.

If the instruction frame includes the instruction code and/or the option code, the user must configure the instruction code and/or the option code to send by writing the fields INST and OPTCODE bit fields in the Instruction Control Register (INSTRCTRL.OPTCODE, INSTRCTRL.INSTR).

Then, the user must write the Instruction Frame Register (INSTRFRAME) to configure the instruction frame depending on which instruction must be sent. If the instruction frame does not include data, writing in this register triggers the send of the instruction frame in the QSPI. If the instruction frame includes data, the send of the instruction frame is triggered by the first data access in the QSPI memory space.

The instruction frame is configured by the following bits and fields of INSTRFRAME:

- WIDTH field is used to configure which data lanes are used to send the instruction code, the address, the option code and to transfer the data. It is possible to use two unidirectional data lanes (MISO-MOSI Single-bit SPI), two bidirectional data lanes (DATA0 - DATA1 Dual SPI) or four bidirectional data lanes (DATA0 - DATA3).

**Table 37-3.** WIDTH Encoding

| INSTRFRAME | Instruction | Address/Option | Data |
|---|---|---|---|
| 0 | Single-bit SPI | Single-bit SPI | Single-bit SPI |
| 1 | Single-bit SPI | Single-bit SPI | Dual SPI |
| 2 | Single-bit SPI | Single-bit SPI | Quad SPI |
| 3 | Single-bit SPI | Dual SPI | Dual SPI |
| 4 | Single-bit SPI | Quad SPI | Quad SPI |
| 5 | Dual SPI | Dual SPI | Dual SPI |
| 6 | Quad SPI | Quad SPI | Quad SPI |
| 7 | Reserved | | |

- INSTREN bit enables sending an instruction code
- ADDREN bit enables sending of an address after the instruction code
- OPTCODEEN bit enables sending of an option code after the address
- DATAEN bit enables the transfer of data (READ or PROGRAM instruction)
- OPTCODELEN field configures the option code length (0 -> 1-bit / 1 -> 2-bit / 2 -> 4-bit / 3 -> 8-bit). The value written in OPTCODELEN must be consistent with value written in the field WIDTH. For


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

example: OPTCODELEN = 0 (1-bit option code) is not coherent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4-bit).

- ADDRLEN bit configures the address length (0 -> 24 bits / 1-> 32 bits)
- TFRTYPE field defines which type of data transfer must be performed
- DUMMYLEN field configures the number of dummy cycles when reading data from the serial Flash memory. Between the address/option and the data, with some instructions, dummy cycles are inserted by the serial Flash memory.

If data transfer is enabled, the user can access the serial memory by reading or writing the QSPI memory space following these rules:

- Reading from the serial memory, but not memory data (for example reading the JEDEC-ID or the STATUS), requires TFRTYPE to be written to 0x0
- Reading from the serial memory, and particularly memory data, requires TFRTYPE to be written to '1'
- Writing to the serial memory, but not memory data (for example writing the configuration or STATUS), requires TFRTYPE to be written to 0x2
- Writing to the serial memory, and particularly memory data, requires TFRTYPE to be written to 0x3

If TFRTYP has a value other than 0x1 and CTRLB.SMEMREG=0, the address sent in the instruction frame is the address of the first system bus accesses. The addresses of the subsequent access actions are not used by the QSPI. At each system bus access, an SPI transfer is performed with the same size. For example, a half-word system bus access leads to a 16-bit SPI transfer, and a byte system bus access leads to an 8-bit SPI transfer.

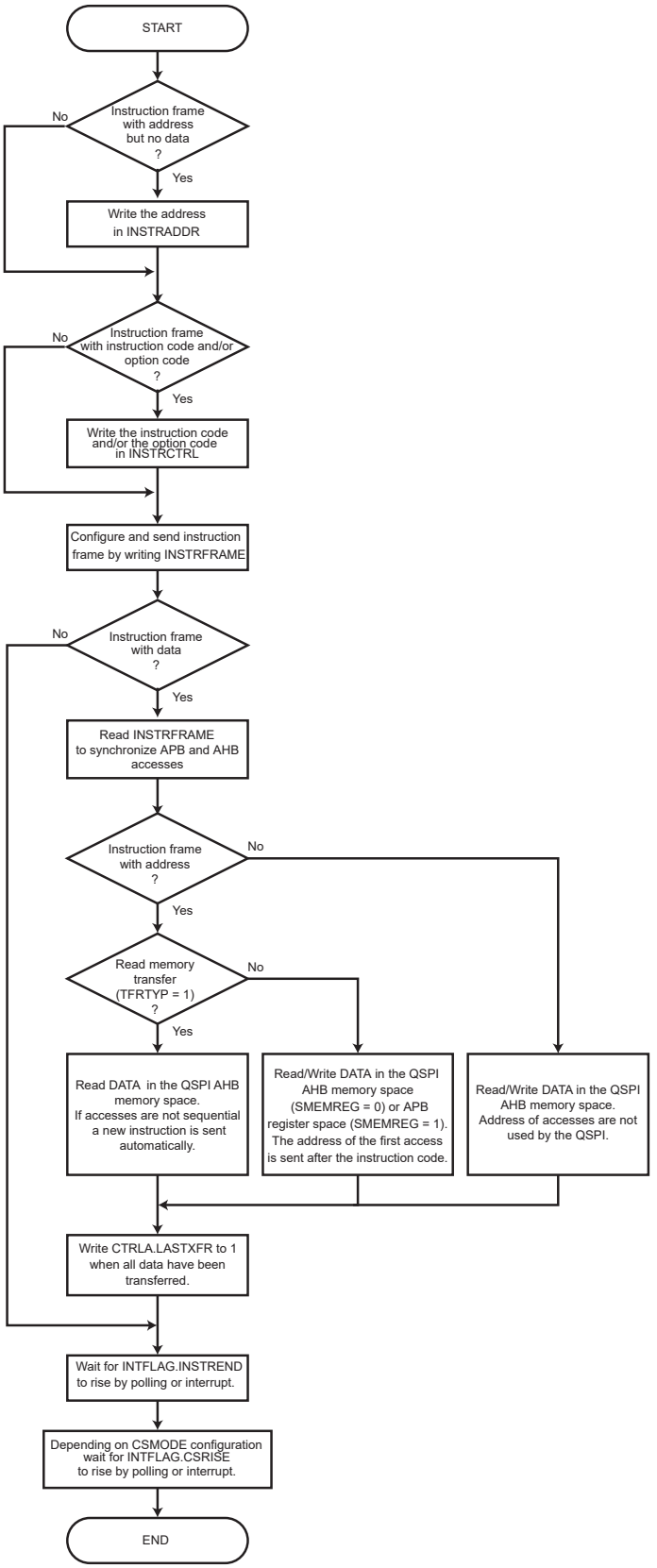
If CTRLB.SMEMREG=1, accesses are made via the QSPI registers and the address sent in the instruction frame is the address defined in the INSTRADDR register. Each time the INSTRFRAME or TXDATA registers are written, an SPI transfer is performed with a byte size. Another byte is read each time RXDATA register is read or written each time TXDATA register is written. The SPI transfer ends by writing the LASTXFER bit in Control A register (CTRLA.LASTXFER).

If TFRTYP=0x1, the address of the first instruction frame is the one of the first read access in the QSPI memory space. Each time the read accesses become non-sequential (addresses are not consecutive), a new instruction frame is sent with the last system bus access address. In this way, the system can read data at a random location in the serial memory. The size of the SPI transfers may differ from the size of the system bus read accesses.

When data transfer is not enabled, the end of the instruction frame is indicated when the INSTREND interrupt flag in the INTFLAG register is set. When data transfer is enabled, the user must indicate when data transfer is completed in the QSPI memory space by setting the bit LASTXFR in the CTRLA. The end of the instruction frame is indicated when the INSTREND interrupt flag in the INTFLAG register is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 37-10.** Instruction Transmission Flow Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.6.8.3. Read Memory Transfer

The user can access the data of the serial memory by sending an instruction with DATAEN=1 and TFRTYP=0x1 in the Instruction Frame register (INSTRFRAME).

In this mode the QSPI is able to read data at random address into the serial Flash memory, allowing the CPU to execute code directly from it (XIP execute-in-place).

In order to fetch data, the user must first configure the instruction frame by writing the INSTRFRAME. Then data can be read at any address in the QSPI address space mapping. The address of the system bus read accesses match the address of the data inside the serial Flash memory.

When Fetch Mode is enabled, several instruction frames can be sent before writing the bit LASTXFR in the CTRLA. Each time the system bus read accesses become non-sequential (addresses are not consecutive), a new instruction frame is sent with the corresponding address.

### 37.6.8.4. Continuous Read Mode

The QSPI is compatible with Continuous Read Mode (CRM) which is implemented in some Serial Flash memories.

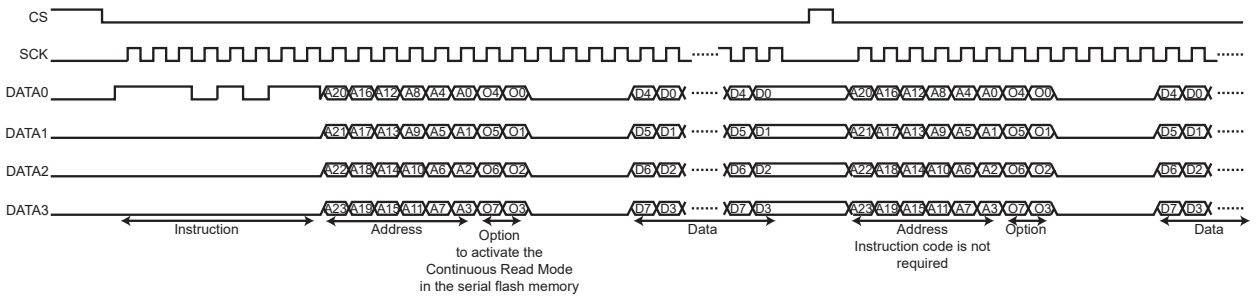
The CRM allows to reduce the instruction overhead by excluding the instruction code from the instruction frame. When CRM is activated in a Serial Flash memory (by a specific option code), the instruction code is stored in the memory. For the next instruction frames, the instruction code is not required, as the memory uses the stored one.

In the QSPI, CRM is used when reading data from the memory (INSTFRAME.TFRTYPE=0x1). The addresses of the system bus read accesses are often non-sequential, this leads to many instruction frames with always the same instruction code. By disabling the sending of the instruction code, the CRM reduces the access time of the data.

To be functional, this mode must be enabled in both the QSPI and the Serial Flash memory. The CRM is enabled in the QSPI by setting the CRM bit in the INSTRFRAME register (INSTFRAME.CRMODE=1, INSTFRAME.TFRTYPE must be 0x1). The CRM is enabled in the Serial Flash memory by sending a specific option code.

**CAUTION** If CRM is not supported by the Serial Flash memory or disabled, the CRMODE bit must not be set. Otherwise, data read out the Serial Flash memory is not valid.

**Figure 37-11.** Continuous Read Mode




### 37.6.8.5. Instruction Frame Transmission Examples

All waveforms in the following examples describe SPI transfers in SPI Clock mode 0 (BAUD.CPOL=0 and BAUD.CPHA=0). All system bus accesses described below refer to the system bus address phase. System bus wait cycles and system bus data phases are not shown.

**Example 37-1.** Example 1

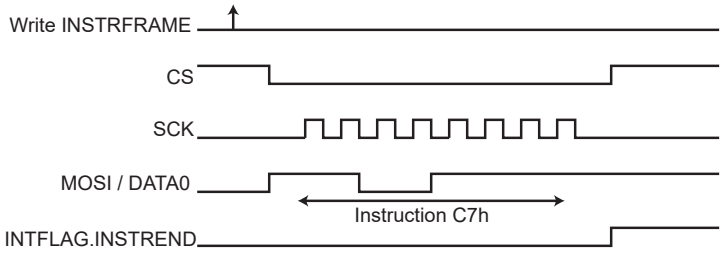
Instruction in Single-bit SPI, without address, without option, without data.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Command: CHIP ERASE (C7h).

- Write 0x0000_00C7 to INSTRCTRL register
- Write 0x0000_0010 to INSTRFRAME register
- Wait for INTFLAG.INSTREND to rise

**Figure 37-12.** Instruction Transmission Waveform 1



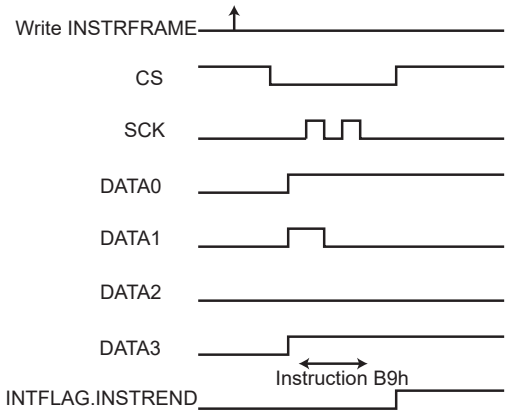

**Example 37-2.** Example 2

Instruction in Quad SPI, without address, without option, without data.

Command: POWER DOWN (B9h)

- Write 0x0000_00B9 to INSTRCTRL register
- Write 0x0000_0016 to INSTRFRAME register
- Wait for INTFLAG.INSTREND to rise

**Figure 37-13.** Instruction Transmission Waveform 2




**Example 37-3.** Example 3

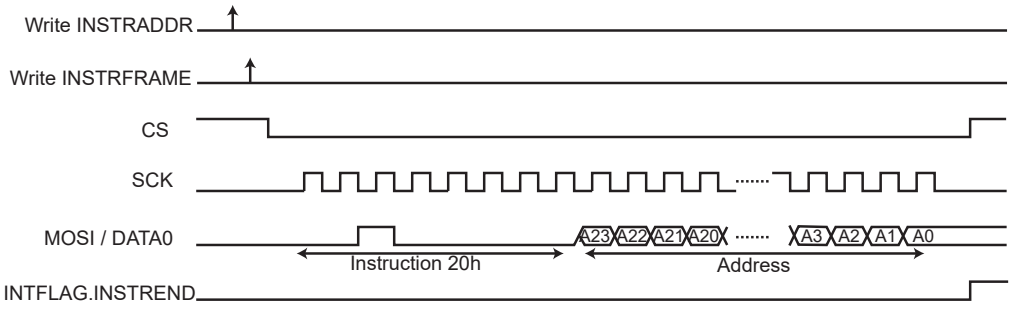
Instruction in Single-bit SPI, with address in Single-bit SPI, without option, without data.

Command: BLOCK ERASE (20h)

- Write the address (of the block to erase) to QSPI_AR
- Write 0x0000_0020 to INSTRCTRL register
- Write 0x0000_0030 toINSTRFRAME register
- Wait for INTFLAG.INSTREND to rise


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 37-14.** Instruction Transmission Waveform 3



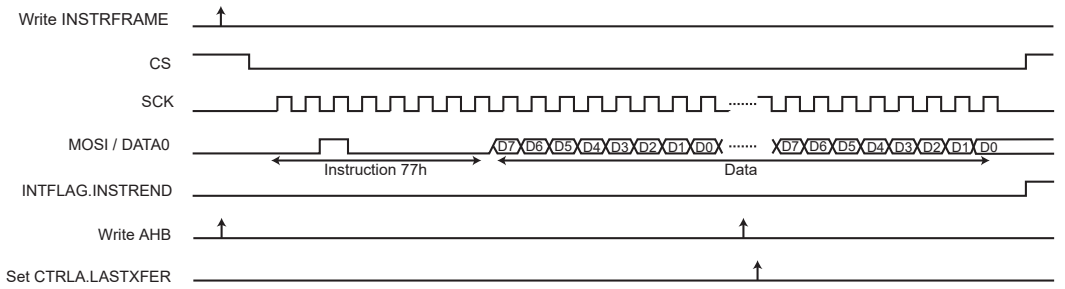

**Example 37-4.** Example 4

Instruction in Single-bit SPI, without address, without option, with data write in Single-bit SPI.

Command: SET BURST (77h)

- Write 0x0000_0077 to INSTRCTRL register.
- Write 0x0000_2090 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Write data to the system bus memory space (0x0400_0000–0x0500_0000). The address of the system bus write accesses is not used.
- Write the LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

**Figure 37-15.** Instruction Transmission Waveform 4




**Example 37-5.** Example 5

Instruction in Single-bit SPI, with address in Dual SPI, without option, with data write in Dual SPI.

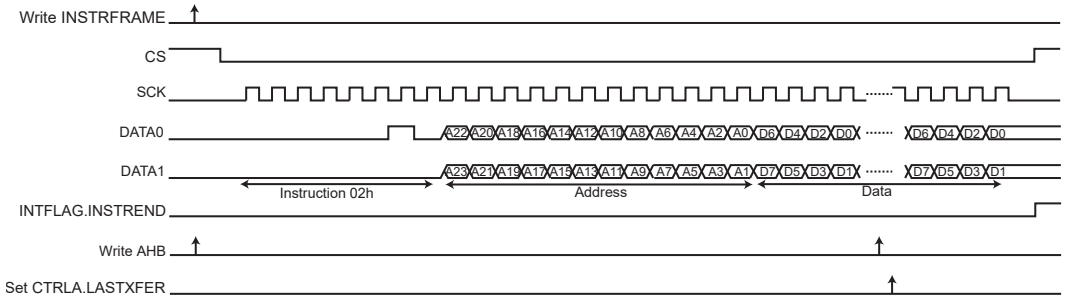
Command: BYTE/PAGE PROGRAM (02h)

- Write 0x0000_0002 to INSTRCTRL register.
- Write 0x0000_30B3 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Write data to the QSPI system bus memory space (0x040 00000–0x0500_0000). The address of the first system bus write access is sent in the instruction frame. The address of the next system bus write accesses is not used.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

**Figure 37-16.** Instruction Transmission Waveform 5



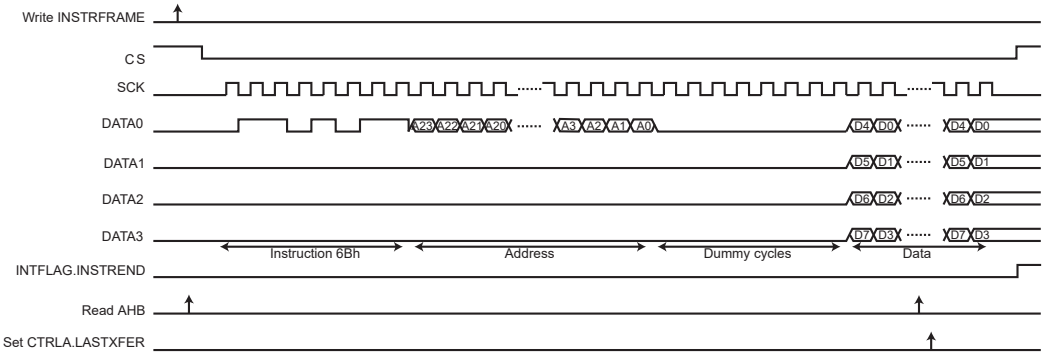

**Example 37-6.** Example 6

Instruction in Single-bit SPI, with address in Single-bit SPI, without option, with data read in Quad SPI, with eight dummy cycles.

Command: QUAD_OUTPUT READ ARRAY (6Bh)

- Write 0x0000_006B to INSTRCTRL register.
- Write 0x0008_10B2 ti INSTRFRAME register.
- Read QSPI_IR (dummy read) to synchronize system bus accesses.
- Read data from the QSPI system bus memory space (0x040 00000–0x0500_0000).
  The address of the first system bus read access is sent in the instruction frame.
  The address of the next system bus read accesses is not used.
- Write the LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

**Figure 37-17.** Instruction Transmission Waveform 6




**Example 37-7.** Example 7

Instruction in Single-bit SPI, with address and option in Quad SPI, with data read from Quad SPI, with four dummy cycles, with fetch and continuous read.

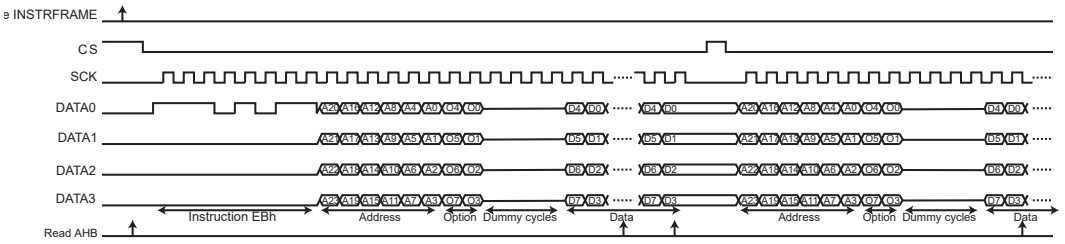
Command: FAST READ QUAD I/O (EBh) - 8-BIT OPTION (0x30h)

- Write 0x0030_00EB to INSTRCTRL register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Write 0x0004_33F4 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Read data from the QSPI system bus memory space (0x040 00000–0x0500_0000). Fetch is enabled, the address of the system bus read accesses is always used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

**Figure 37-18.** Instruction Transmission Waveform 7



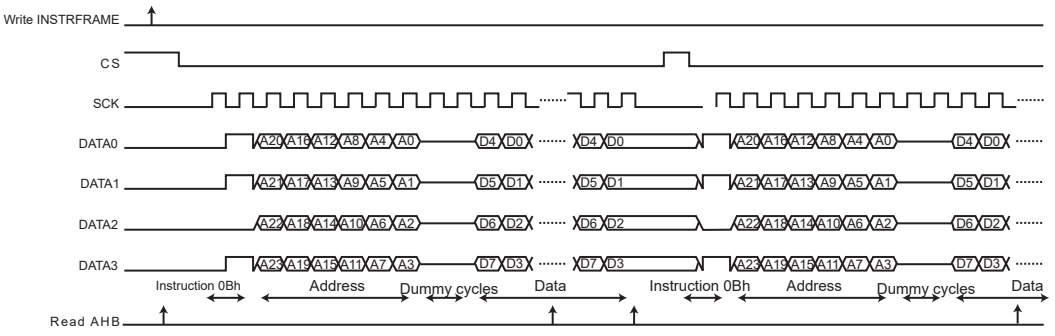

**Example 37-8.** Example 8

Instruction in Quad SPI, with address in Quad SPI, without option, with data read from Quad SPI, with two dummy cycles, with fetch.

Command: HIGH-SPEED READ (0Bh)

- Write 0x0000_000B to INSTRCTRL register.
- Write 0x0002_20B6 to INSTRFRAME register.
- Read INSTRFRAME register (dummy read) to synchronize system bus accesses.
- Read data in the QSPI system bus memory space (0x040 00000–0x0500_0000). Fetch is enabled, the address of the system bus read accesses is always used.
- Write LASTXFR bit in CTRLA register to '1'.
- Wait for INTFLAG.INSTREND to rise.

**Figure 37-19.** Instruction Transmission Waveform 8




## 37.6.9. Scrambling/Unscrambling Function

The scrambling/unscrambling function cannot be performed on devices other than memories. Data is scrambled when written to memory and unscrambled when data is read.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The external data lines can be scrambled in order to prevent intellectual property data located in off-chip memories from being easily recovered by analyzing data at the package pin level of either the micro-controller or the QSPI Client device (e.g. memory).

The scrambling/unscrambling function can be enabled by writing a '1' to the ENABLE bit in the Scrambling Control register (SCRAMBCTRL.ENABLE).

The scrambling and unscrambling are performed on-the-fly without impacting the throughput.

The scrambling method depends on the user-configurable Scrambling User Key in the Scrambling Key register (SCRAMBKEY.KEY). This register is only accessible in write mode.

By default, the scrambling and unscrambling algorithm includes the scrambling user key, plus a device-dependent random value. This random value is not included when the Scrambling/Unscrambling Random Value Disable bit in the Scrambling Mode register (SCRAMBCTRL.RANDOMDIS) is written to ‘1’.

The random value is neither user configurable nor readable. If SCRAMBCTRL.RANDOMDIS=0, data scrambled by a given circuit cannot be unscrambled by a different circuit.

If SCRAMBCTRL.RANDOMDIS=1, the scrambling/unscrambling algorithm includes only the scrambling user key, making it possible to manage data by different circuits. Note that the same key must be used by the different circuits.

The scrambling user key must be securely stored in a reliable non-volatile memory in order to recover data from the off-chip memory. Any data scrambled with a given key cannot be recovered if the key is lost.

### 37.6.10. DMA Operation

The QSPI generates the following DMA requests:

- Data received (RX): The request is set when data is available in the RXDATA register, and cleared when RXDATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TXDATA) is empty, and cleared when TXDATA is written.

**Note:** If DMA and RX memory modes are selected, a QSPI memory space read operation is required to force the first triggering.

If the CPU accesses the registers which are source of DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted.

### 37.6.11. Interrupts

The QSPI has the following interrupt source:

- Interrupt Request (INTREQ): Indicates that at least one bit in the Interrupt Flag Status and Clear register (INTFLAG) is set to '1'.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the QSPI is reset. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 37.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | LASTXFER |
| 0x04 | CTRLB | 7:0 | | | CSMODE[1:0] | | SMEMREG | WDRBT | LOOPEN | MODE |
| | | 15:8 | | | | | DATALEN[3:0] | | | |
| | | 23:16 | DLYBCT[7:0] | | | | | | | |
| | | 31:24 | DLYCS[7:0] | | | | | | | |
| 0x08 | BAUD | 7:0 | | | | | | | CPHA | CPOL |
| | | 15:8 | BAUD[7:0] | | | | | | | |
| | | 23:16 | DLYBS[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | RXDATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | TXDATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | | | | | ERROR | TXC | DRE | RXC |
| | | 15:8 | | | | | | INSTREND | | CSRISE |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | INTENSET | 7:0 | | | | | ERROR | TXC | DRE | RXC |
| | | 15:8 | | | | | | INSTREND | | CSRISE |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | INTFLAG | 7:0 | | | | | ERROR | TXC | DRE | RXC |
| | | 15:8 | | | | | | INSTREND | | CSRISE |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | STATUS | 7:0 | | | | | | | ENABLE | |
| | | 15:8 | | | | | | | CSSTATUS | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | INSTRADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x34 | INSTRCTRL | 7:0 | INSTR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | OPTCODE[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x38 | INSTRFRAME | 7:0 | DATAEN | OPTCODEEN | ADDREN | INSTREN | | WIDTH[2:0] | | |
| | | 15:8 | DDREN | CRMODE | TFRTYPE[1:0] | | | ADDRLEN | OPTCODELEN[1:0] | |
| | | 23:16 | | | | DUMMYLEN[4:0] | | | | |
| | | 31:24 | | | | | | | | |
| 0x3C ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | SCRAMBCTRL | 7:0 | | | | | | | RANDOMDIS | ENABLE |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x44 | SCRAMBKEY | 7:0 | KEY[7:0] | | | | | | | |
| | | 15:8 | KEY[15:8] | | | | | | | |
| | | 23:16 | KEY[23:16] | | | | | | | |
| | | 31:24 | KEY[31:24] | | | | | | | |

## 37.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Refer to the Peripheral Access Controller for more information.

Some registers are enable-protected, meaning they can only be written when the QSPI is disabled. Enable-protection is denoted by the Enable-Protected property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** -

Control A

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | LASTXFER |
| Access | | | | | | | | W |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | W | W |
| Reset | | | | | | | 0 | 0 |

**Bit 24 – LASTXFER** Last Transfer
0: No effect.
1: The chip select will be de-asserted after the character written in TD has been transferred.

**Bit 1 – ENABLE** Enable
Writing a '0' to this bit disables the QSPI.
Writing a '1' to this bit enables the QSPI to transfer and receive data.
As soon as ENABLE is reset, QSPI finishes its transfer.
All pins are set in input mode and no data is received or transmitted.
If a transfer is in progress, the transfer is finished before the QSPI is disable.

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets the QSPI. A software-triggered hardware reset of the QSPI interface is performed.
DMAC channels are not affected by software reset.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection

Control B

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DLYCS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DLYBCT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DATALEN[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | CSMODE[1:0] | | SMEMREG | WDRBT | LOOPEN | MODE |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:24 – DLYCS[7:0]** Minimum Inactive CS Delay
This bit field defines the minimum delay between the inactivation and the activation of CS. The DLYCS time guarantees the Client minimum deselect time.
If DLYCS is 0x00, one CLK_QSPI_AHB period will be inserted by default.
Otherwise, the following equation determines the delay:
DLYCS = Minimum inactive × fperipheral clock

**Bits 23:16 – DLYBCT[7:0]** Delay Between Consecutive Transfers
This field defines the delay between two consecutive transfers with the same peripheral without removing the chip select. The delay is always inserted after each transfer and before removing the chip select if needed.
When DLYBCT=0x00, no delay between consecutive transfers is inserted and the clock keeps its duty cycle over the character transfers. In Serial Memory mode (MODE=1), DLYBCT is ignored and no delay is inserted. Otherwise, the following equation determines the delay:
DLYBCT = (Delay Between Consecutive Transfers × fperipheral clock) / 32

**Bits 11:8 – DATALEN[3:0]** Data Length
The DATALEN field determines the number of data bits transferred. Reserved values should not be used.

| Value | Name | Description |
|---|---|---|
| 0x0 | 8BITS | 8-bits transfer |
| 0x1 | 9BITS | 9-bits transfer |
| 0x2 | 10BITS | 10-bits transfer |
| 0x3 | 11BITS | 11-bits transfer |
| 0x4 | 12BITS | 12-bits transfer |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x5 | 13BITS | 13-bits transfer |
| 0x6 | 14BITS | 14-bits transfer |
| 0x7 | 15BITS | 15-bits transfer |
| 0x8 | 16BITS | 16-bits transfer |
| 0x9-0xF | | Reserved |

**Bits 5:4 – CSMODE[1:0]** Chip Select Mode

The CSMODE field determines how the chip select is de-asserted.

| Value | Name | Description |
|---|---|---|
| 0x0 | NORELOAD | The chip select is de-asserted if TD has not been reloaded before the end of the current transfer. |
| 0x1 | LASTXFER | The chip select is de-asserted when the bit LASTXFER is written at 1 and the character written in TD has been transferred. |
| 0x2 | SYSTEMATICALLY | The chip select is de-asserted systematically after each transfer. |
| 0x3 | | Reserved |

**Bit 3 – SMEMREG** Serial Memory Register Mode

| Value | Description |
|---|---|
| 0 | Serial memory registers are written via AHB access. |
| 1 | Serial memory registers are written via APB access. Reset the QSPI. |

**Bit 2 – WDRBT** Wait Data Read Before Transfer

This bit determines the Wait Data Read Before Transfer option.

**Bit 1 – LOOPEN** Local Loopback Enable

This bit defines if the Local Loopback is enabled or disabled.
LOOPEN controls the local loopback on the data serializer for testing in SPI Mode only. (MISO is internally connected on MOSI).

| Value | Description |
|---|---|
| 0 | Local Loopback is disabled. |
| 1 | Local Loopback is enabled. |

**Bit 0 – MODE** Serial Memory Mode

This bit defines if the QSPI is in SPI Mode or Serial Memory Mode.

| Value | Name | Description |
|---|---|---|
| 0 | SPI | SPI operating mode |
| 1 | MEMORY | Serial Memory operating mode |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.3. Baud Rate

**Name:** BAUD
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DLYBS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BAUD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CPHA | CPOL |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 23:16 – DLYBS[7:0]** Delay Before SCK
This field defines the delay from CS valid to the first valid SCK transition.
When DLYBS equals zero, the CS valid to SCK transition is 1/2 the SCK clock period.
Otherwise, the following equation determines the delay:

**Equation 37-1.** Delay Before SCK

$$Delay\ \ Before\ SCK = \frac{DLYBS}{MCK}$$

**Bits 15:8 – BAUD[7:0]** Serial Clock Baud Rate
The QSPI uses a modulus counter to derive the SCK baud rate from the module clock CLK_QSPI_AHB. The Baud rate is selected by writing a value from 0 to 255 in the BAUD field. The following equation determines the SCK baud rate:

**Equation 37-2.** SCK Baud Rate

$$SCK\ \ Baud\ \ Rate = \frac{MCK}{(BAUD + 1)}$$

**Bit 1 – CPHA** Clock Phase
CPHA determines which edge of SCK causes data to change and which edge causes data to be captured. CPHA is used with CPOL to produce the required clock/data relationship between Host and Client devices.

| Value | Description |
|---|---|
| 0 | Data is captured on the leading edge of SCK and changed on the following edge of SCK. |
| 1 | Data is changed on the leading edge of SCK and captured on the following edge of SCK. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – CPOL** Clock Polarity

CPOL is used to determine the inactive state value of the serial clock (SCK). It is used with CPHA to produce the required clock/data relationship between Host and Client devices.

| Value | Description |
|---|---|
| 0 | The inactive state value of SCK is logic level zero. |
| 0 | The inactive state value of SCK is logic level 'one'. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 37.8.4. Receive Data

**Name:** RXDATA
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DATA[15:0]** Receive Data
Data received by the QSPI is stored in this register right-justified. Unused bits read zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.5. Transmit Data

**Name:** TXDATA
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DATA[15:0]** Transmit Data
Data to be transmitted by the QSPI is stored in this register. Information to be transmitted must be written to the transmit data register in a right-justified format.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.6. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | INSTREND | | CSRISE |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ERROR | TXC | DRE | RXC |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 10 – INSTREND** Instruction End Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' will clear the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The INSTREND interrupt is disabled. |
| 1 | The INSTREND interrupt is enabled. |

**Bit 8 – CSRISE** Chip Select Rise Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' will clear the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The CSRISE interrupt is disabled. |
| 1 | The CSRISE interrupt is enabled. |

**Bit 3 – ERROR** Overrun Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' will clear the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The ERROR interrupt is disabled. |
| 1 | The ERROR interrupt is enabled. |

**Bit 2 – TXC** Transmission Complete Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' will clear the corresponding interrupt request.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The TXC interrupt is disabled. |
| 1 | The TXC interrupt is enabled. |

**Bit 1 – DRE** Transmit Data Register Empty Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' will clear the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The DRE interrupt is disabled. |
| 1 | The DRE interrupt is enabled. |

**Bit 0 – RXC** Receive Data Register Full Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' will clear the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The RXC interrupt is disabled. |
| 1 | The RXC interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 37.8.7. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | INSTREND | | CSRISE |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ERROR | TXC | DRE | RXC |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 10 – INSTREND** Instruction End Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' will set the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The INSTREND interrupt is disabled. |
| 1 | The INSTREND interrupt is enabled. |

**Bit 8 – CSRISE** Chip Select Rise Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' will set the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The CSRISE interrupt is disabled. |
| 1 | The CSRISE interrupt is enabled. |

**Bit 3 – ERROR** Overrun Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' will set the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The ERROR interrupt is disabled. |
| 1 | The ERROR interrupt is enabled. |

**Bit 2 – TXC** Transmission Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' will set the corresponding interrupt request.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The TXC interrupt is disabled. |
| 1 | The TXC interrupt is enabled. |

**Bit 1 – DRE** Transmit Data Register Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' will set the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The DRE interrupt is disabled. |
| 1 | The DRE interrupt is enabled. |

**Bit 0 – RXC** Receive Data Register Full Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' will set the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The RXC interrupt is disabled. |
| 1 | The RXC interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 37.8.8. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | INSTREND | | CSRISE |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ERROR | TXC | DRE | RXC |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 10 – INSTREND** Instruction End
This bit is set when an Instruction End has been detected.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 8 – CSRISE** Chip Select Rise
The bit is set when a Chip Select Rise has been detected.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 3 – ERROR** Overrun Error
This bit is set when an ERROR has occurred.
An ERROR occurs when RXDATA is loaded at least twice from the serializer.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the flag.

**Bit 2 – TXC** Transmission Complete
0: As soon as data is written in TXDATA.
1: TXDATA and internal shifter are empty. If a transfer delay has been defined, TXC is set after the completion of such delay.

**Bit 1 – DRE** Transmit Data Register Empty
0: Data has been written to TXDATA and not yet transferred to the serializer.
1: The last data written in the TXDATA has been transferred to the serializer.
This bit is '0' when the QSPI is disabled or at reset.
The bit is set as soon as ENABLE bit is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – RXC** Receive Data Register Full

0: No data has been received since the last read of RXDATA.

1: Data has been received and the received data has been transferred from the serializer to RXDATA since the last read of RXDATA.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.9. Status

**Name:** STATUS
**Offset:** 0x20
**Reset:** 0x00000200
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CSSTATUS | |
| Access | | | | | | | R | |
| Reset | | | | | | | 1 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | |
| Access | | | | | | | R | |
| Reset | | | | | | | 0 | |

**Bit 9 – CSSTATUS** Chip Select

| Value | Description |
|---|---|
| 0 | Chip Select is asserted. |
| 1 | Chip Select is not asserted. |

**Bit 1 – ENABLE** Enable

| Value | Description |
|---|---|
| 0 | QSPI is disabled. |
| 1 | QSPI is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.10. Instruction Address

**Name:** INSTRADDR
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADDR[31:0]** Instruction Address
Address to send to the serial Flash memory in the instruction frame.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.11. Instruction Code

**Name:** INSTRCTRL
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | OPTCODE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | INSTR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – OPTCODE[7:0]** Option Code
These bits define the option code to send to the serial flash memory.

**Bits 7:0 – INSTR[7:0]** Instruction Code
Instruction code to send to the serial flash memory.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.12. Instruction Frame

**Name:** INSTRFRAME
**Offset:** 0x38
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | DUMMYLEN[4:0] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DDREN | CRMODE | TFRTYPE[1:0] | | | ADDRLEN | OPTCODELEN[1:0] | |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATAEN | OPTCODEEN | ADDREN | INSTREN | | WIDTH[2:0] | | |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 20:16 – DUMMYLEN[4:0]** Dummy Cycles Length
The DUMMYLEN field defines the number of dummy cycles required by the serial Flash memory before data transfer.

**Bit 15 – DDREN** Double Data Rate Enable
**Note:** Double Data Rate operating is only supported in Read.

| Value | Description |
|---|---|
| 0 | Double Data Rate operating mode is disabled. |
| 1 | Double Data Rate operating mode is enabled. |

**Bit 14 – CRMODE** Continuous Read Mode
This bit defines if the Continuous Read Mode is enabled or disabled.

| Value | Description |
|---|---|
| 0 | Continuous Read Mode is disabled. |
| 1 | Continuous Read Mode is enabled. |

**Bits 13:12 – TFRTYPE[1:0]** Data Transfer Type
These bits define the data type transfer.

| Value | Name | Description |
|---|---|---|
| 0x0 | READ | Read transfer from the serial memory.Scrambling is not performed.Read at random location (fetch) in the serial flash memory is not possible. |
| 0x1 | READMEMORY | Read data transfer from the serial memory.If enabled, scrambling is performed.Read at random location (fetch) in the serial flash memory is possible. |
| 0x2 | WRITE | Write transfer into the serial memory.Scrambling is not performed. |
| 0x3 | WRITEMEMORY | Write data transfer into the serial memory. If enabled, scrambling is performed. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 10 – ADDRLEN** Address Length
The ADDRLEN bit determines the length of the address.

| Value | Name | Description |
|---|---|---|
| 0x0 | 24BITS | 24-bits address length |
| 0x1 | 32BITS | 32-bits address length |

**Bits 9:8 – OPTCODELEN[1:0]** Option Code Length
The OPTCODELEN field determines the length of the option code. The value written in OPTCODELEN must be coherent with value written in the field WIDTH. For example: OPTCODELEN=0 (1-bit option code) is not coherent with WIDTH=6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4-bit).

| Value | Name | Description |
|---|---|---|
| 0x0 | 1BIT | 1-bit length option code |
| 0x1 | 2BITS | 2-bits length option code |
| 0x2 | 4BITS | 4-bits length option code |
| 0x3 | 8BITS | 8-bits length option code |

**Bit 7 – DATAEN** Data Enable

| Value | Description |
|---|---|
| 0 | No data is sent/received to/from the serial flash memory. |
| 1 | Data is sent/received to/from the serial flash memory. |

**Bit 6 – OPTCODEEN** Option Enable

| Value | Description |
|---|---|
| 0 | The option is not sent to the serial flash memory |
| 1 | The option is sent to the serial flash memory. |

**Bit 5 – ADDREN** Address Enable

| Value | Description |
|---|---|
| 0 | The transfer address is not sent to the serial flash memory. |
| 1 | The transfer address is sent to the serial flash memory. |

**Bit 4 – INSTREN** Instruction Enable

| Value | Description |
|---|---|
| 0 | The instruction is not sent to the serial flash memory. |
| 1 | The instruction is sent to the serial flash memory. |

**Bits 2:0 – WIDTH[2:0]** Instruction Code, Address, Option Code and Data Width
This field defines the width of the instruction code, the address, the option and the data.

| Value | Name | Description |
|---|---|---|
| 0x0 | SINGLE_BIT_SPI | Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Single-bit SPI |
| 0x1 | DUAL_OUTPUT | Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Dual SPI |
| 0x2 | QUAD_OUTPUT | Instruction: Single-bit SPI / Address-Option: Single-bit SPI / Data: Quad SPI |
| 0x3 | DUAL_IO | Instruction: Single-bit SPI / Address-Option: Dual SPI / Data: Dual SPI |
| 0x4 | QUAD_IO | Instruction: Single-bit SPI / Address-Option: Quad SPI / Data: Quad SPI |
| 0x5 | DUAL_CMD | Instruction: Dual SPI / Address-Option: Dual SPI / Data: Dual SPI |
| 0x6 | QUAD_CMD | Instruction: Quad SPI / Address-Option: Quad SPI / Data: Quad SPI |
| 0x7 | | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.13. Scrambling Mode

**Name:** SCRAMBCTRL
**Offset:** 0x40
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RANDOMDIS | ENABLE |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – RANDOMDIS** Scrambling/Unscrambling Random Value Disable

| Value | Description |
|---|---|
| 0 | The scrambling/unscrambling algorithm includes the scrambling user key plus a random value that may differ from chip to chip. |
| 1 | The scrambling/unscrambling algorithm includes only the scrambling user key. |

**Bit 0 – ENABLE** Scrambling/Unscrambling Enable
This bit defines if the scrambling/unscrambling is enabled or disabled.

| Value | Description |
|---|---|
| 0 | Scrambling/unscrambling is disabled. |
| 1 | Scrambling/unscrambling is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.8.14. Scrambling Key

**Name:** SCRAMBKEY
**Offset:** 0x44
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | KEY[31:24] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | KEY[23:16] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | KEY[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | KEY[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – KEY[31:0]** Scrambling User Key
This field defines the user key value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 38. USB – Universal Serial Bus

## 38.1. Overview

The Universal Serial Bus interface (USB) module complies with the Universal Serial Bus (USB) 2.1 specification supporting both device and embedded host modes.

The USB device mode supports 8 endpoint addresses. All endpoint addresses have one input and one output endpoint, for a total of 16 endpoints. Each endpoint is fully configurable in any of the four transfer types: control, interrupt, bulk or isochronous. The USB host mode supports up to 8 pipes. The maximum data payload size is selectable up to 1023 bytes.

Internal SRAM is used to keep the configuration and data buffer for each endpoint. The memory locations used for the endpoint configurations and data buffers is fully configurable. The amount of memory allocated is dynamic according to the number of endpoints in use, and the configuration of these. The USB module has a built-in Direct Memory Access (DMA) and will read/write data from/to the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

To maximize throughput, an endpoint can be configured for ping-pong operation. When this is done the input and output endpoint with the same address are used in the same direction. The CPU or DMA Controller can then read/write one data buffer while the USB module writes/reads from the other buffer. This gives double buffered communication.

Multi-packet transfer enables a data payload exceeding the maximum packet size of an endpoint to be transferred as multiple packets without any software intervention. This reduces the number of interrupts and software intervention needed for USB transfers.

For low power operation the USB module can put the microcontroller in any sleep mode when the USB bus is idle and a suspend condition is given. Upon bus resume, the USB module can wake the microcontroller from any sleep mode.

## 38.2. Features

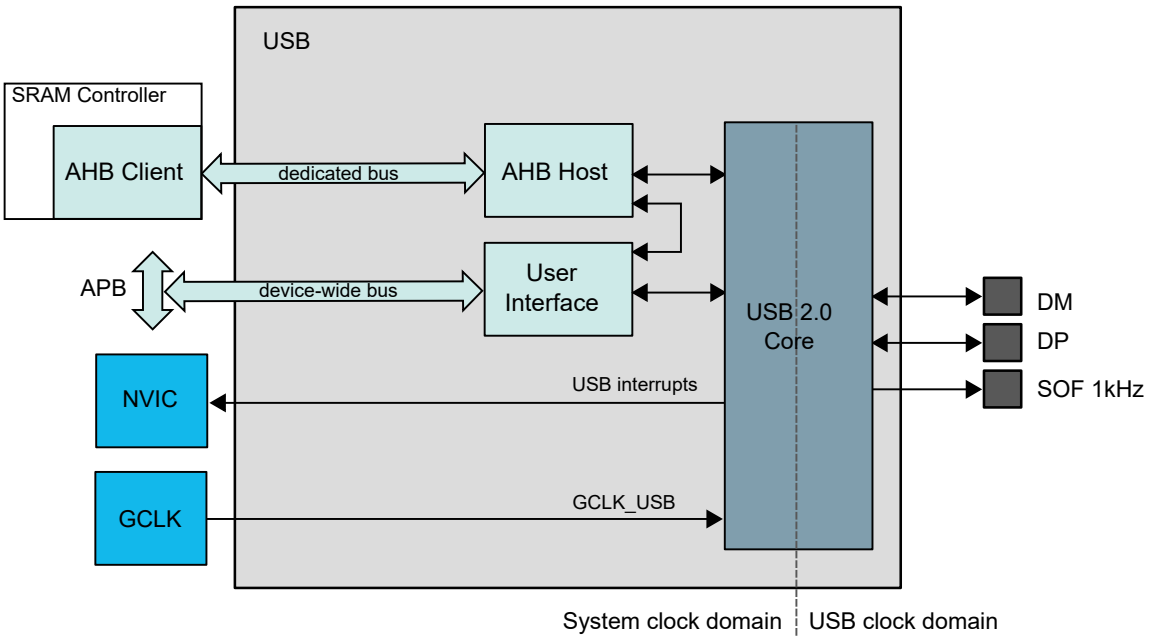
- Compatible with the USB 2.1 specification
- USB Embedded Host and Device mode
- Supports full (12Mbit/s) and low (1.5Mbit/s) speed communication
- Supports Link Power Management (LPM-L1) protocol
- On-chip transceivers with built-in pull-ups and pull-downs
- On-Chip USB serial resistors
- 1kHz SOF clock available on external pin
- Device mode
  - Supports 8 IN endpoints and 8 OUT endpoints
  - No endpoint size limitations
  - Built-in DMA with multi-packet and dual bank for all endpoints
  - Supports feedback endpoint
  - Supports crystal less clock
- Host mode
  - Supports 8 physical pipes
  - No pipe size limitations
  - Supports multiplexed virtual pipe on one physical pipe to allow an unlimited USB tree
  - Built-in DMA with multi-packet support and dual bank for all pipes


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Supports feedback endpoint
- Supports the USB 2.0 Phase-locked SOFs feature

## 38.3. USB Block Diagram

**Figure 38-1.** LS/FS Implementation: USB Block Diagram




## 38.4. Signal Description

| Pin Name | Pin Description | Type |
|---|---|---|
| DM | Data -: Differential Data Line - Port | Input/Output |
| DP | Data +: Differential Data Line + Port | Input/Output |
| SOF 1kHZ | SOF Output | Output |

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

**Related Links**

I/O Multiplexing and Considerations

## 38.5. Product Dependencies

In order to use this peripheral module, other parts of the system must be configured correctly, as described below.

### 38.5.1. I/O Lines

The USB pins may be multiplexed with the I/O lines Controller. The user must first configure the I/O Controller to assign the USB pins to their peripheral functions.

A 1kHz SOF clock is available on an external pin. The user must first configure the I/O Controller to assign the 1kHz SOF clock to the peripheral function. The SOF clock is available for device and host mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake up the device from Sleep modes. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

**Related Links**

PM – Power Manager

### 38.5.3. Clocks

The USB bus clock (CLK_USB_AHB) can be enabled and disabled in the Main Clock module, MCLK, and the default state of CLK_USB_AHB can be found in the *Peripheral Clock Masking*.

A generic clock (GCLK_USB) is required to clock the USB. This clock must be configured and enabled in the Generic Clock Controller before using the USB.

This generic clock is asynchronous to the bus clock (CLK_USB_AHB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

The USB module requires a GCLK_USB of 48 MHz ± 0.25% clock for low speed and full speed operation. To follow the USB data rate at 12 Mbit/s in full-speed mode, the CLK_USB_AHB clock should be at minimum 8 MHz. The GCLK_USB clock is generated by the DFLL48 using a reference clock. When the USB is disabled, the GCLK used as DFLL reference should be disabled.

Clock recovery is achieved by a digital phase-locked loop in the USB module, which complies with the USB jitter specifications. If crystal-less operation is used in USB device mode, refer to *USB Clock Recovery Module*.

**Related Links**

GCLK - Generic Clock Controller
Synchronization
AHBMASK
AHB Mask
Additional Features

### 38.5.4. DMA

The USB has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a USB transaction takes place. No CPU or DMA Controller resources are required.

### 38.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller

### 38.5.6. Events

Not applicable.

### 38.5.7. Debug Operation

When the CPU is halted in debug mode the USB peripheral continues normal operation. If the USB peripheral is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.5.8. Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Device Interrupt Flag (INTFLAG) register
- Endpoint Interrupt Flag (EPINTFLAG) register
- Host Interrupt Flag (INTFLAG) register
- Pipe Interrupt Flag (PINTFLAG) register

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

### 38.5.9. Analog Connections

Not applicable.

### 38.5.10. Calibration

The output drivers for the DP/DM USB line interface can be fine tuned with calibration values from production tests. The calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register (PADCAL) by software, before enabling the USB, to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.

For details on Pad Calibration, refer to the Pad Calibration (PADCAL) register.

## 38.6. Functional Description

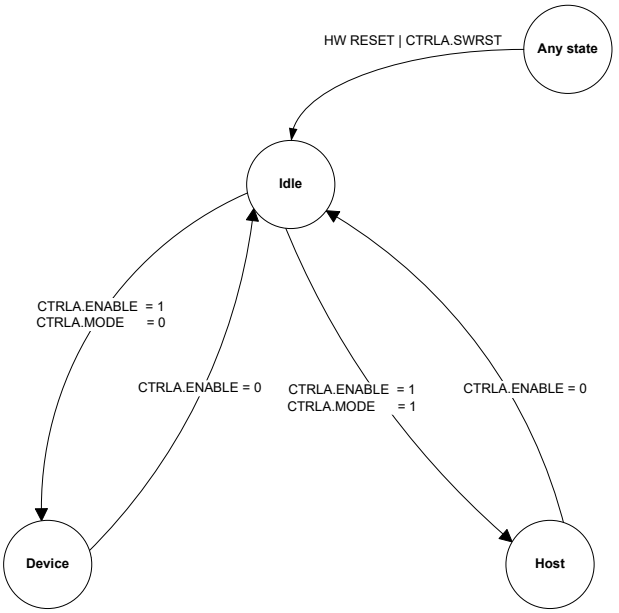
### 38.6.1. USB General Operation

#### 38.6.1.1. Initialization

After a hardware reset, the USB is disabled. The user should first enable the USB (CTRLA.ENABLE) in either device mode or host mode (CTRLA.MODE).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-2.** General States




After a hardware reset, the USB is in the idle state. In this state:

- The module is disabled. The USB Enable bit in the Control A register (CTRLA.ENABLE) is reset.
- The module clock is stopped in order to minimize power consumption.
- The USB pad is in suspend mode.
- The internal states and registers of the device and host are reset.

Before using the USB, the Pad Calibration register (PADCAL) must be loaded with production calibration values from the NVM Software Calibration Area.

The USB is enabled by writing a '1' to CTRLA.ENABLE. The USB is disabled by writing a '0' to CTRLA.ENABLE.

The USB is reset by writing a '1' to the Software Reset bit in CTRLA (CTRLA.SWRST). All registers in the USB will be reset to their initial state, and the USB will be disabled. Refer to the CTRLA register for details.

The user can configure pads and speed before enabling the USB by writing to the Operating Mode bit in the Control A register (CTRLA.MODE) and the Speed Configuration field in the Control B register (CTRLB.SPDCONF). These values are taken into account once the USB has been enabled by writing a '1' to CTRLA.ENABLE.

After writing a '1' to CTRLA.ENABLE, the USB enters device mode or host mode (according to CTRLA.MODE).

The USB can be disabled at any time by writing a '0' to CTRLA.ENABLE.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Refer to USB Device Operations for the basic operation of the device mode.

Refer to Host Operations for the basic operation of the host mode.

### 38.6.2. USB Device Operations

This section gives an overview of the USB module device operation during normal transactions. For more details on general USB and USB protocol, refer to the Universal Serial Bus specification revision 2.1.

#### 38.6.2.1. Initialization

To attach the USB device to start the USB communications from the USB host, a zero should be written to the Detach bit in the Device Control B register (CTRLB.DETACH). To detach the device from the USB host, a one must be written to the CTRLB.DETACH.

After the device is attached, the host will request the USB device descriptor using the default device address zero. On successful transmission, it will send a USB reset. After that, it sends an address to be configured for the device. All further transactions will be directed to this device address. This address should be configured in the Device Address field in the Device Address register (DADD.DADD) and the Address Enable bit in DADD (DADD.ADDEN) should be written to one to accept communications directed to this address. DADD.ADDEN is automatically cleared on receiving a USB reset.

#### 38.6.2.2. Endpoint Configuration

Endpoint data can be placed anywhere in the device RAM. The USB controller accesses these endpoints directly through the AHB host (built-in DMA) with the help of the endpoint descriptors. The base address of the endpoint descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. For additional information, refer to Endpoint Description Structure.

Before using an endpoint, the user should configure the direction and type of the endpoint in Type of Endpoint field in the Device Endpoint Configuration register (EPCFG.EPTYPE0/1). The endpoint descriptor registers should be initialized to known values before using the endpoint, so that the USB controller does not read random values from the RAM.

The Endpoint Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported to the host for that endpoint. The Address of Data Buffer register (ADDR) should be set to the data buffer used for endpoint transfers.

The RAM Access Interrupt bit in Device Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during IN data stage.

When an endpoint is disabled, the following registers are cleared for that endpoint:

- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)

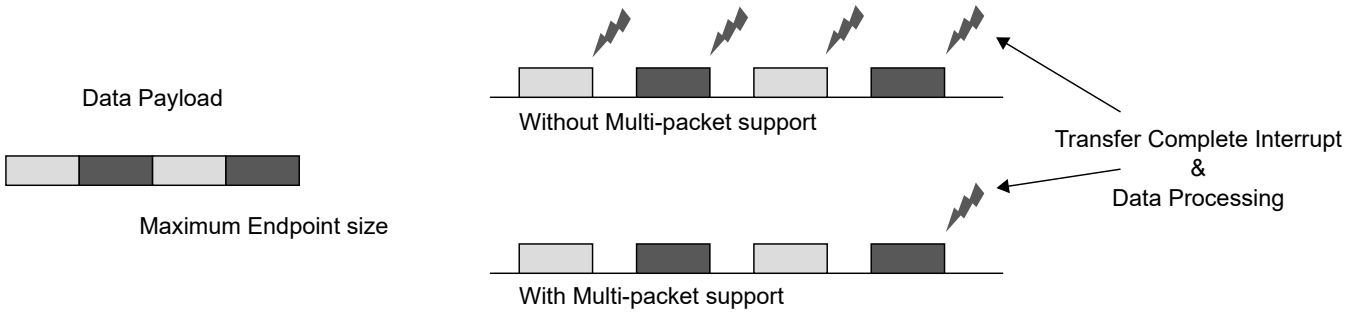
#### 38.6.2.3. Multi-Packet Transfers

Multi-packet transfer enables a data payload exceeding the endpoint maximum transfer size to be transferred as multiple packets without software intervention. This reduces the number of interrupts and software intervention required to manage higher level USB transfers. Multi-packet transfer is identical to the IN and OUT transactions described below unless otherwise noted in this section.

The application software provides the size and address of the RAM buffer to be proceeded by the USB module for a specific endpoint, and the USB module will split the buffer in the required USB data transfers without any software intervention.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-3.** Multi-Packet Feature - Reduction of CPU Overhead




#### 38.6.2.4. USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- Device Frame Number (FNUM) register
- Device Address (DADD) register
- Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- Device Endpoint Interrupt Flag (EPINTFLAG) register
- Transmit Stall 0 bit in the Endpoint Status register (EPSTATUS.STALLRQ0)
- Transmit Stall 1 bit in the Endpoint Status register (EPSTATUS.STALLRQ1)
- Endpoint Interrupt Summary (EPINTSMRY) register
- Upstream resume bit in the Control B register (CTRLB.UPRSM)

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

#### 38.6.2.5. Start-of-Frame

When a Start-of-Frame (SOF) token is detected, the frame number from the token is stored in the Frame Number field in the Device Frame Number register (FNUM.FNUM), and the Start-of-Frame interrupt bit in the Device Interrupt Flag register (INTFLAG.SOF) is set. If there is a CRC or bit-stuff error, the Frame Number Error status flag (FNUM.FNCERR) in the FNUM register is set.

#### 38.6.2.6. Management of SETUP Transactions

When a SETUP token is detected and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint is enabled in EPCFG. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed endpoint. If the EPCFG.EPTYPE0 is not set to control, the USB module returns to idle and waits for the next token packet.

When the EPCFG.EPTYPE0 matches, the USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor and waits for a DATA0 packet. If a PID error or any other PID than DATA0 is detected, the USB module returns to idle and waits for the next token packet.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

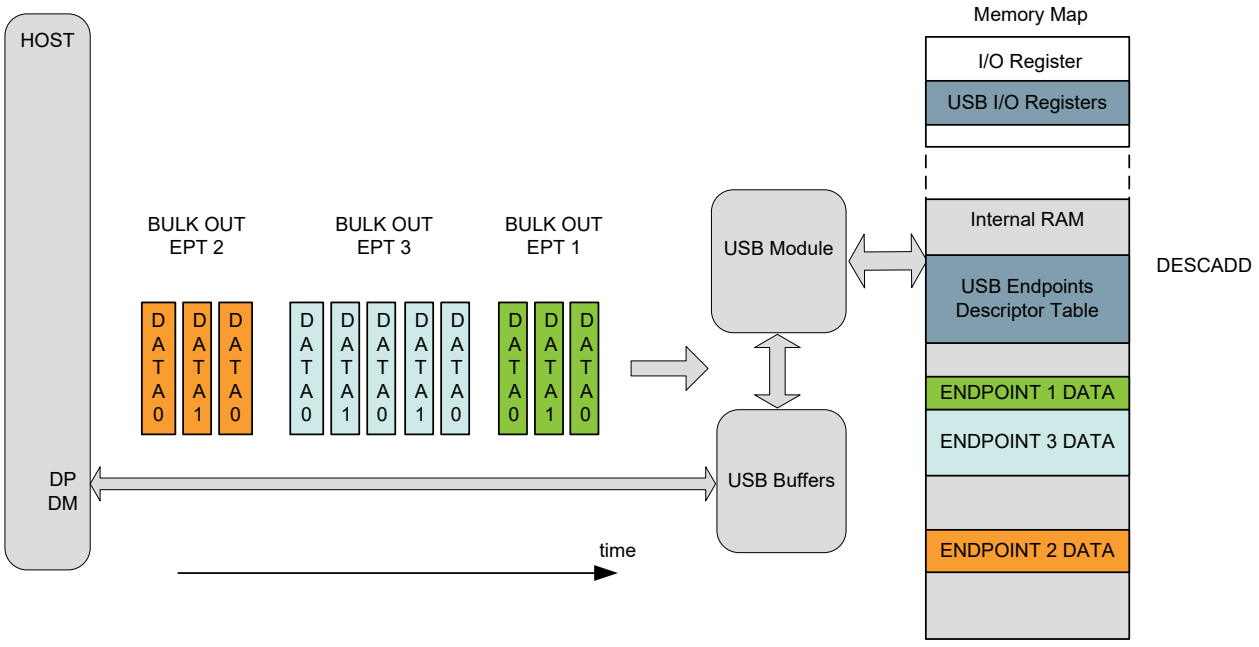
When the data PID matches and if the Received Setup Complete interrupt bit in the Device Endpoint Interrupt Flag register (EPINTFLAG.RXSTP) is equal to zero, ignoring the Bank 0 Ready bit in the Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the endpoint's maximum data payload size as specified by the PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. Software must never report a endpoint size to the host that is greater than the value configured in PCKSIZE.SIZE. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If data is successfully received, an ACK handshake is returned to the host, and the number of received data bytes, excluding the CRC, is written to the Byte Count (PCKSIZE.BYTE_COUNT). If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE, no CRC data is written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data is written to the data buffer. If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally the EPSTATUS is updated. Data Toggle OUT bit (EPSTATUS.DTGLOUT), the Data Toggle IN bit (EPSTATUS.DTGLIN), the current bank bit (EPSTATUS.CURRBK) and the Bank Ready 0 bit (EPSTATUS.BK0RDY) are set. Bank Ready 1 bit (EPSTATUS.BK1RDY) and the Stall Bank 0/1 bit (EPSTATUS.STALLQR0/1) are cleared on receiving the SETUP request. The RXSTP bit is set and triggers an interrupt if the Received Setup Interrupt Enable bit is set in Endpoint Interrupt Enable Set/Clear register (EPINTENSET/CLR.RXSTP).

### 38.6.2.7. Management of OUT Transactions

**Figure 38-4.** OUT Transfer: Data Packet Host to USB Device




When an OUT token is detected, and the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

If the address matches, the USB module checks if the endpoint number received is enabled in the EPCFG of the addressed endpoint. If the addressed endpoint is disabled, the packet is discarded and the USB module returns to idle and waits for the next token packet.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When the endpoint is enabled, the USB module then checks the Endpoint Configuration register (EPCFG) of the addressed output endpoint. If the type of the endpoint (EPCFG.EPTYPE0) is not set to OUT, the USB module returns to idle and waits for the next token packet.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor, and waits for a DATA0 or DATA1 packet. If a PID error or any other PID than DATA0 or DATA1 is detected, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ0 in EPSTATUS is set, the incoming data is discarded. If the endpoint is not isochronous, a STALL handshake is returned to the host and the Transmit Stall Bank 0 interrupt bit in EPINTFLAG (EPINTFLAG.STALL0) is set.

For isochronous endpoints, data from both a DATA0 and DATA1 packet will be accepted. For other endpoint types the PID is checked against EPSTATUS.DTGLOUT. If a PID mismatch occurs, the incoming data is discarded, and an ACK handshake is returned to the host.

If EPSTATUS.BK0RDY is set, the incoming data is discarded, the bit Transmit Fail 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRFAIL0) and the status bit STATUS_BK.ERRORFLOW are set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the maximum data payload specified as PCKSIZE.SIZE, the remainders of the received data bytes are discarded. The packet will still be checked for bit-stuff and CRC errors. If a bit-stuff or CRC error is detected in the packet, the USB module returns to idle and waits for the next token packet.

If the endpoint is isochronous and a bit-stuff or CRC error in the incoming data, the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. Finally the EPINTFLAG.TRFAIL0 and CRC Error bit in the Device Bank Status register (STATUS_BK.CRCERR) is set for the addressed endpoint.

If data was successfully received, an ACK handshake is returned to the host if the endpoint is not isochronous, and the number of received data bytes, excluding CRC, is written to PCKSIZE.BYTE_COUNT. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE no CRC data bytes are written to the data buffer. If the number of received data bytes is the maximum data payload specified by PCKSIZE.SIZE minus one, only the first CRC data byte is written to the data buffer If the number of received data is equal or less than the data payload specified by PCKSIZE.SIZE minus two, both CRC data bytes are written to the data buffer.

Finally in EPSTATUS for the addressed output endpoint, EPSTATUS.BK0RDY is set and EPSTATUS.DTGLOUT is toggled if the endpoint is not isochronous. The flag Transmit Complete 0 interrupt bit in EPINTFLAG (EPINTFLAG.TRCPT0) is set for the addressed endpoint.

#### 38.6.2.8. Multi-Packet Transfers for OUT Endpoint

The number of data bytes received is stored in endpoint PCKSIZE.BYTE_COUNT as for normal operation. Since PCKSIZE.BYTE_COUNT is updated after each transaction, it must be set to zero when setting up a new transfer. The total number of bytes to be received must be written to PCKSIZE.MULTI_PACKET_SIZE. This value must be a multiple of PCKSIZE.SIZE, otherwise excess data may be written to SRAM locations used by other parts of the application.

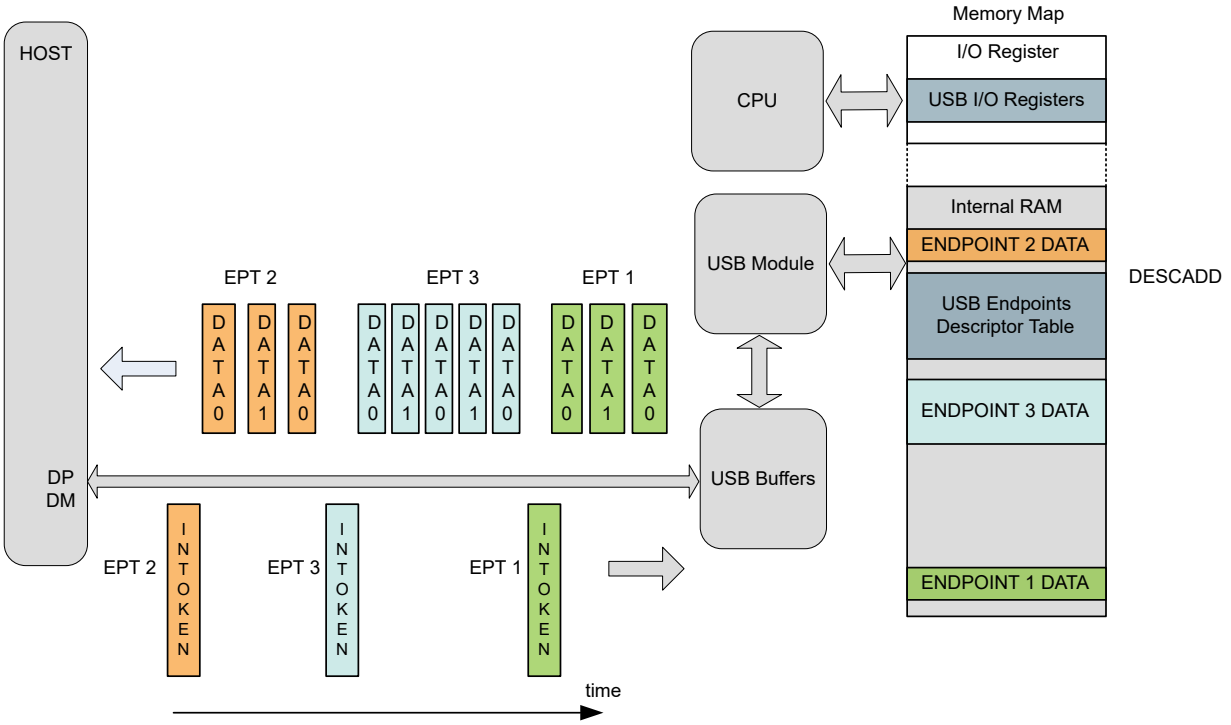
EPSTATUS.DTGLOUT management for non-isochronous packets and EPINTFLAG.BK1RDY/BK0RDY management are as for normal operation.

If a maximum payload size packet is received, PCKSIZE.BYTE_COUNT will be incremented by PCKSIZE.SIZE after the transaction has completed, and EPSTATUS.DTGLOUT will be toggled if the endpoint is not isochronous. If the updated PCKSIZE.BYTE_COUNT is equal to PCKSIZE.MULTI_PACKET_SIZE (i.e. the last transaction), EPSTATUS.BK1RDY/BK0RDY, and EPINTFLAG.TRCPT0/TRCPT1 will be set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.6.2.9. Management of IN Transactions

**Figure 38-5.** IN Transfer: Data Packet USB Device to Host After Request from Host




When an IN token is detected, and if the device address of the token packet does not match DADD.DADD, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the address matches, the USB module checks if the endpoint received is enabled in the EPCFG of the addressed endpoint and if not, the packet is discarded and the USB module returns to idle and waits for the next token packet.

When the endpoint is enabled, the USB module then checks on the EPCFG of the addressed input endpoint. If the EPCFG.EPTYPE1 is not set to IN, the USB module returns to idle and waits for the next token packet.

If EPSTATUS.STALLRQ1 in EPSTATUS is set, and the endpoint is not isochronous, a STALL handshake is returned to the host and EPINTFLAG.STALL1 is set.

If EPSTATUS.BK1RDY is cleared, the flag EPINTFLAG.TRFAIL1 is set. If the endpoint is not isochronous, a NAK handshake is returned to the host.

The USB module then fetches the Data Buffer Address (ADDR) from the addressed endpoint's descriptor. The data pointed to by the Data Buffer Address (ADDR) is sent to the host in a DATA0 packet if the endpoint is isochronous. For non-isochronous endpoints a DATA0 or DATA1 packet is sent depending on the state of EPSTATUS.DTGLIN. When the number of data bytes specified in endpoint PCKSIZE.BYTE_COUNT is sent, the CRC is appended and sent to the host.

For isochronous endpoints, EPSTATUS.BK1RDY is cleared and EPINTFLAG.TRCPT1 is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For all non-isochronous endpoints the USB module waits for an ACK handshake from the host. If an ACK handshake is not received within 16 bit times, the USB module returns to idle and waits for the next token packet. If an ACK handshake is successfully received EPSTATUS.BK1RDY is cleared, EPINTFLAG.TRCPT1 is set and EPSTATUS.DTGLIN is toggled.

### 38.6.2.10. Multi-Packet Transfers for IN Endpoint

The total number of data bytes to be sent is written to PCKSIZE.BYTE_COUNT as for normal operation. The Multi-packet size register (PCKSIZE.MULTI_PACKET_SIZE) is used to store the number of bytes that are sent, and must be written to zero when setting up a new transfer.

When an IN token is received, PCKSIZE.BYTE_COUNT and PCKSIZE.MULTI_PACKET_SIZE are fetched. If PCKSIZE.BYTE_COUNT minus PCKSIZE.MULTI_PACKET_SIZE is less than the endpoint PCKSIZE.SIZE, endpoint BYTE_COUNT minus endpoint PCKSIZE.MULTI_PACKET_SIZE bytes are transmitted, otherwise PCKSIZE.SIZE number of bytes are transmitted. If endpoint PCKSIZE.BYTE_COUNT is a multiple of PCKSIZE.SIZE, the last packet sent will be zero-length if the AUTOZLP bit is set.

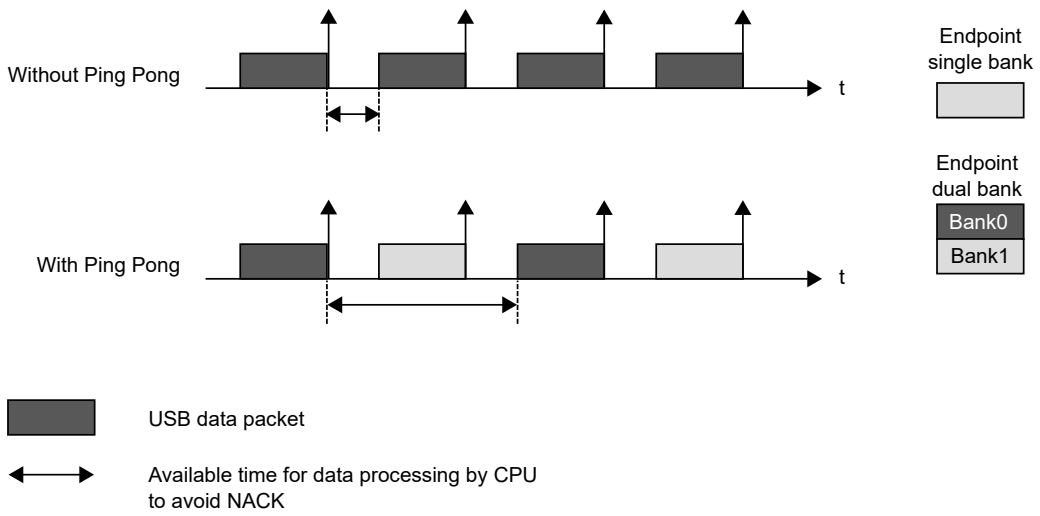
If a maximum payload size packet was sent (i.e. not the last transaction), MULTI_PACKET_SIZE will be incremented by the PCKSIZE.SIZE. If the endpoint is not isochronous the EPSTATUS.DTLGIN bit will be toggled when the transaction has completed. If a short packet was sent (i.e. the last transaction), MULTI_PACKET_SIZE is incremented by the data payload. EPSTATUS.BK0/1RDY will be cleared and EPINTFLAG.TRCPT0/1 will be set.

### 38.6.2.11. Ping-Pong Operation

When an endpoint is configured for ping-pong operation, it uses both the input and output data buffers (banks) for a given endpoint in a single direction. The direction is selected by enabling one of the IN or OUT direction in EPCFG.EPTYPE0/1 and configuring the opposite direction in EPCFG.EPTYPE1/0 as Dual Bank.

When ping-pong operation is enabled for an endpoint, the endpoint in the opposite direction must be configured as dual bank. The data buffer, data address pointer and byte counter from the enabled endpoint are used as Bank 0, while the matching registers from the disabled endpoint are used as Bank 1.

**Figure 38-6.** Ping-Pong Overview




The Bank Select flag in EPSTATUS.CURBK indicates which bank data will be used in the next transaction, and is updated after each transaction. According to EPSTATUS.CURBK, EPINTFLAG.TRCPT0 or EPINTFLAG.TRFAIL0 or EPINTFLAG.TRCPT1 or EPINTFLAG.TRFAIL1 in


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

EPINTFLAG and Data Buffer 0/1 ready (EPSTATUS.BK0RDY and EPSTATUS.BK1RDY) are set. The EPSTATUS.DTGLOUT and EPSTATUS.DTGLIN are updated for the enabled endpoint direction only.

### 38.6.2.12. Feedback Operation

Feedback endpoints are endpoints with same the address but in different directions. This is usually used in explicit feedback mechanism in USB Audio, where a feedback endpoint is associated to one or more isochronous data endpoints to which it provides feedback service. The feedback endpoint always has the opposite direction from the data endpoint.

The feedback endpoint always has the opposite direction from the data endpoint(s). The feedback endpoint has the same endpoint number as the first (lower) data endpoint. A feedback endpoint can be created by configuring an endpoint with different endpoint size (PCKSIZE.SIZE) and different endpoint type (EPCFG.EPTYPE0/1) for the IN and OUT direction.

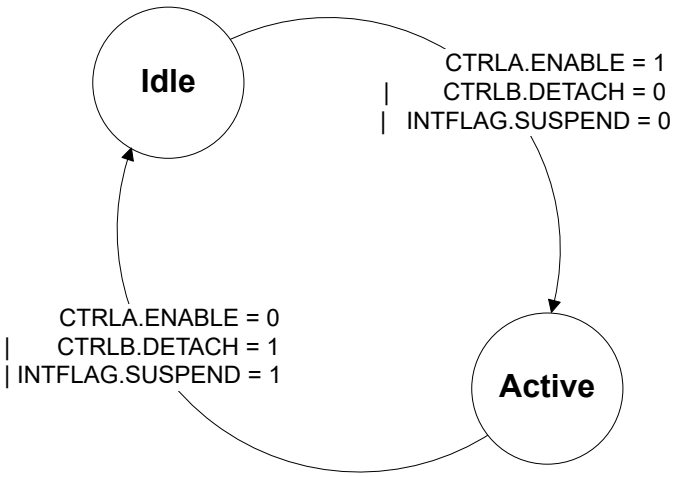
Example Configuration for Feedback Operation:

- Endpoint n / IN: EPCFG.EPTYPE1 = Interrupt IN, PCKSIZE.SIZE = 64.
- Endpoint n / OUT: EPCFG.EPTYPE0= Isochronous OUT, PCKSIZE.SIZE = 512.

### 38.6.2.13. Suspend State and Pad Behavior

The following figure, Pad Behavior, illustrates the behavior of the USB pad in Device mode.

**Figure 38-7.** Pad Behavior



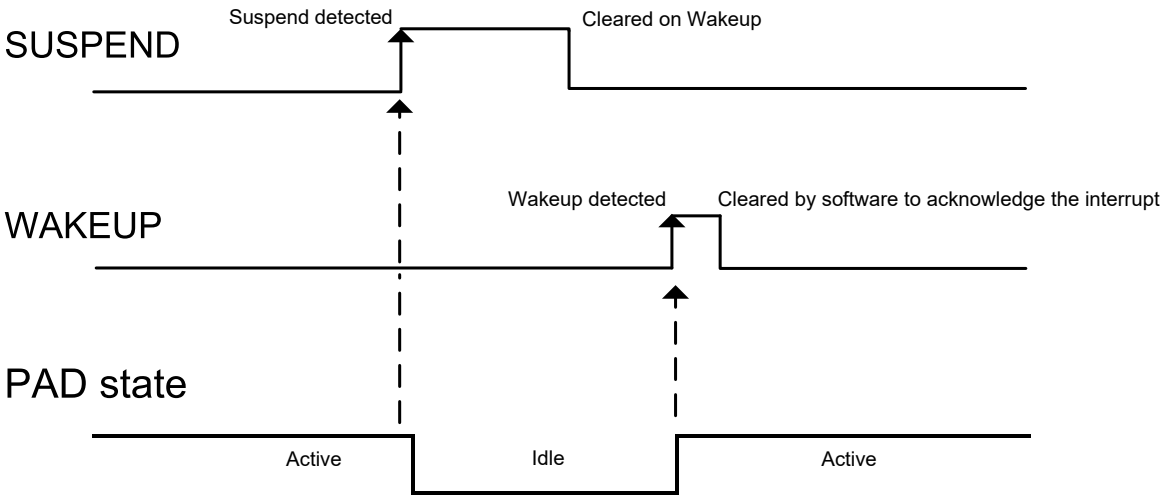

In Idle state, the pad is in Low Power Consumption mode.

In Active state, the pad is active.

The following figure, Pad Events, illustrates the pad events leading to a PAD state change.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-8.** Pad Events




The Suspend Interrupt bit in the Device Interrupt Flag register (INTFLAG.SUSPEND) is set when a USB Suspend state has been detected on the USB bus. The USB pad is then automatically put in the Idle state. The detection of a non-idle state sets the Wake Up Interrupt bit (INTFLAG.WAKEUP) and wakes the USB pad.

The pad goes to the Idle state if the USB module is disabled or if CTRLB.DETACH is written to one. It returns to the Active state when CTRLA.ENABLE is written to one and CTRLB.DETACH is written to zero.

#### 38.6.2.14. Remote Wakeup

The remote wakeup request (also known as upstream resume) is the only request the device may send on its own initiative. This should be preceded by a DEVICE_REMOTE_WAKEUP request from the host.

First, the USB must have detected a “Suspend” state on the bus, i.e. the remote wakeup request can only be sent after INTFLAG.SUSPEND has been set.

The user may then write a one to the Remote Wakeup bit (CTRLB.UPRSM) to send an Upstream Resume to the host initiating the wakeup. This will automatically be done by the controller after 5 ms of inactivity on the USB bus.

When the controller sends the Upstream Resume INTFLAG.WAKEUP is set and INTFLAG.SUSPEND is cleared.

The CTRLB.UPRSM is cleared at the end of the transmitting Upstream Resume.

In case of a rebroadcast resume initiated by the host, the End of Resume bit (INTFLAG.EORSM) flag is set when the rebroadcast resume is completed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

In the case where the CTRLB.UPRSM bit is set while a host initiated downstream resume is already started, the CTRLB.UPRSM is cleared and the upstream resume request is ignored.

#### 38.6.2.15. Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Device

The LPM Handshake bit in CTRLB.LPMHDSK should be configured to accept the LPM transaction.

When a LPM transaction is received on any enabled endpoint n and a handshake has been sent in response by the controller according to CTRLB.LPMHDSK, the Device Link Power Manager (EXTREG) register is updated in the bank 0 of the addressed endpoint's descriptor. It contains information such as the Best Effort Service Latency (BESL), the Remote Wake bit (bRemoteWake), and the Link State parameter (bLinkState). Usually, the LPM transaction uses only the endpoint number 0.

If the LPM transaction was positively acknowledged (ACK handshake), USB sets the Link Power Management Interrupt bit (INTFLAG.LPMSUSP) bit which indicates that the USB transceiver is suspended, reducing power consumption. This suspend occurs 9 microseconds after the LPM transaction according to the specification.

To further reduce consumption, it is recommended to stop the USB clock while the device is suspended.

The MCU can also enter in one of the available sleep modes if the wakeup time latency of the selected sleep mode complies with the host latency constraint (refer to the BESL parameter in the EXTREG register).

Recovering from this LPM-L1 suspend state is exactly the same as the Suspend state (see Section Suspend State and Pad Behavior) except that the remote wakeup duration initiated by USB is shorter to comply with the Link Power Management specification.

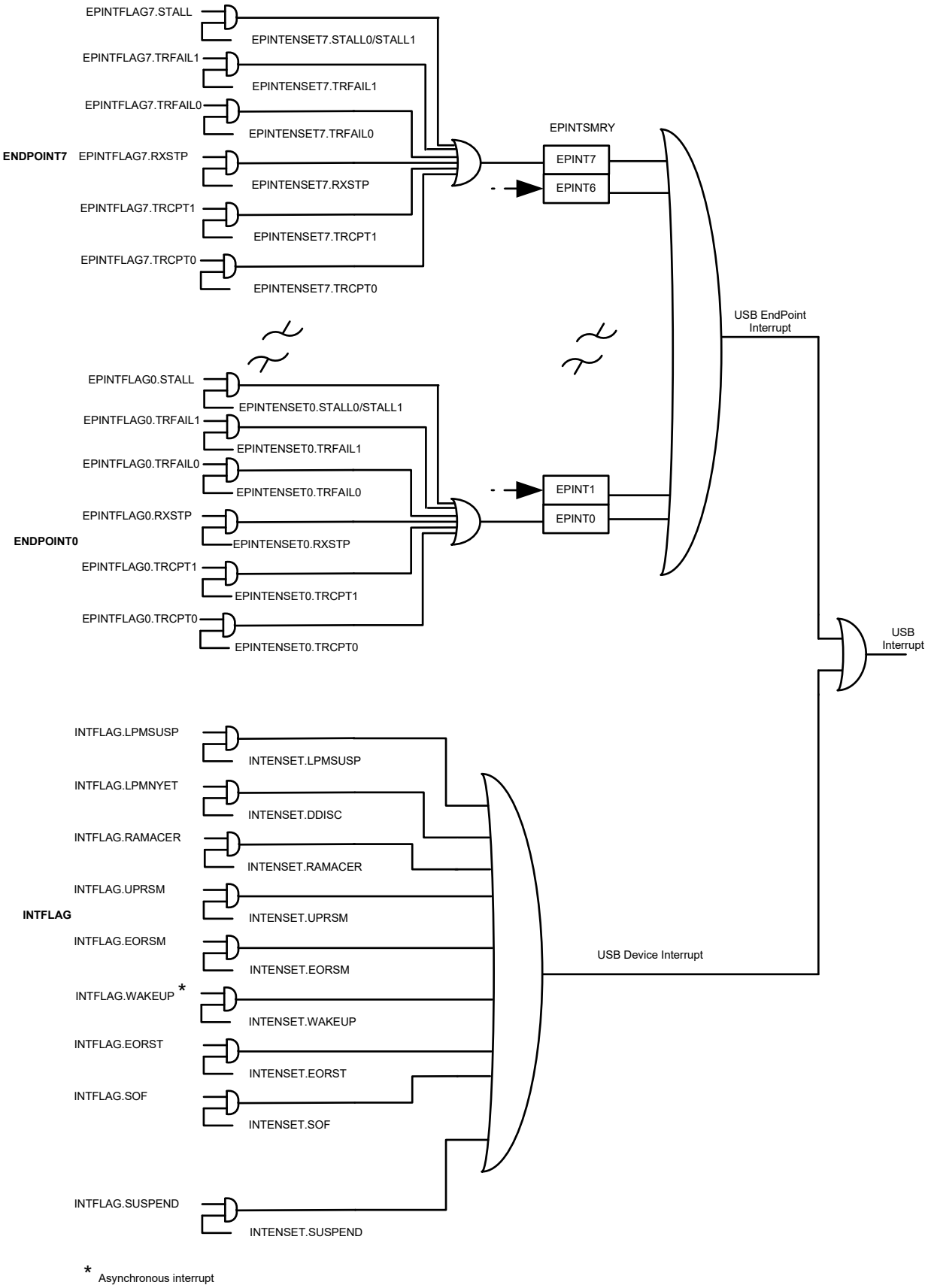
If the LPM transaction is responded with a NYET, the Link Power Management Not Yet Interrupt Flag (INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit (INTENCLR/SET.LPMNYET) is set.

If the LPM transaction is responded with a STALL or no handshake, no flag is set, and the transaction is ignored.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.6.2.16. USB Device Interrupt

**Figure 38-9.** Device Interrupt





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.

### 38.6.3. Host Operations

This section gives an overview of the USB module Host operation during normal transactions. For more details on general USB and USB protocol, refer to Universal Serial Bus Specification revision 2.1.

#### 38.6.3.1. Device Detection and Disconnection

Prior to device detection the software must set the VBUS is OK bit (CTRLB.VBUSOK) register when the VBUS is available. This notifies the USB host that USB operations can be started. When the bit CTRLB.VBUSOK is zero and even if the USB HOST is configured and enabled, host operation is halted. Setting the bit CTRLB.VBUSOK will allow host operation when the USB is configured.

The Device detection is managed by the software using the Line State field in the Host Status (STATUS.LINESTATE) register. The device connection is detected by the host controller when DP or DM is pulled high, depending of the speed of the device.

The device disconnection is detected by the host controller when both DP and DM are pulled down using the STATUS.LINESTATE registers.

The Device Connection Interrupt bit (INTFLAG.DCONN) is set if a device connection is detected.

The Device Disconnection Interrupt bit (INTFLAG.DDISC) is set if a device disconnection is detected.

#### 38.6.3.2. Host Terminology

In host mode, the term pipe is used instead of endpoint. A host pipe corresponds to a device endpoint, refer to "Universal Serial Bus Specification revision 2.1." for more information.

#### 38.6.3.3. USB Reset

The USB sends a USB reset signal when the user writes a one to the USB Reset bit (CTRLB.BUSRESET). When the USB reset has been sent, the USB Reset Sent Interrupt bit in the INTFLAG (INTFLAG.RST) is set and all pipes will be disabled.

If the bus was previously in a suspended state (i.e., the Start of Frame Generation Enable bit (CTRLB.SOFE) is zero), the USB will switch it to the Resume state, causing the bus to asynchronously set the Host Wakeup Interrupt flag (INTFLAG.WAKEUP). The CTRLB.SOFE bit will be set in order to generate SOFs immediately after the USB reset.

During USB reset the following registers are cleared:

- All Host Pipe Configuration register (PCFG)
- Host Frame Number register (FNUM)
- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Host Start-of-Frame Control register (HSOFC)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

After the reset the user should check the Speed Status field in the Status register (STATUS.SPEED) to find out the current speed according to the capability of the peripheral.

#### 38.6.3.4. Pipe Configuration

Pipe data can be placed anywhere in the RAM. The USB controller accesses these pipes directly through the AHB host (built-in DMA) with the help of the pipe descriptors. The base address of the pipe descriptors needs to be written in the Descriptor Address register (DESCADD) by the user. Refer to Pipe Description Structure.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Before using a pipe, the user should configure the direction and type of the pipe in Type of Pipe field in the Host Pipe Configuration register (PCFG.PTYPE). The pipe descriptor registers should be initialized to known values before using the pipe, so that the USB controller does not read the random values from the RAM.

The Pipe Size field in the Packet Size register (PCKSIZE.SIZE) should be configured as per the size reported by the device for the endpoint associated with this pipe. The Address of Data Buffer register (ADDR) should be set to the data buffer used for pipe transfers.

The Pipe Bank bit (PCFG.BK) should be set to one if dual banking is desired. Dual bank is not supported for Control pipes.

The Ram Access Interrupt bit in Host Interrupt Flag register (INTFLAG.RAMACER) is set when a RAM access underflow error occurs during an OUT stage.

When a pipe is disabled, the following registers are cleared for that pipe:

- Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- Pipe Interrupt Flag register (PINTFLAG)
- Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

#### 38.6.3.5. Pipe Activation

A disabled pipe is inactive, and will be reset along with its context registers (pipe registers for the pipe n). Pipes are enabled by writing the Type of the Pipe bit (PCFG.PTYPE) to a value different than 0x0 (disabled).

When a pipe is enabled, the Pipe Freeze bit in the Pipe Status register (PSTATUS.FREEZE) is set. This allows the user to complete the configuration of the pipe, without starting a USB transfer.

When starting an enumeration, the user retrieves the device descriptor by sending a GET_DESCRIPTOR USB request. This descriptor contains the maximal packet size of the device default control endpoint (bMaxPacketSize0), which the user should use to reconfigure the size of the default control pipe.

#### 38.6.3.6. Pipe Address Setup

Once the device has answered the first host requests with the default device address 0, the host assigns a new address to the device. The host controller has to send a USB reset to the device and a SET_ADDRESS(addr) SETUP request with the new address to be used by the device. Once this SETUP transaction is complete, the user writes the new address to the Pipe Device Address field in the Host Control Pipe register (CTRL_PIPE.PDADDR) in Pipe descriptor. All following requests by this pipe will be performed using this new address.

#### 38.6.3.7. Suspend and Wakeup

Setting CTRLB.SOFE to zero when in host mode will cause the USB to cease sending Start-of-Frames on the USB bus and enter the Suspend state. The USB device will enter the Suspend state 3ms later.

Before entering suspend by writing CTRLB.SOFE to zero, the user must freeze the active pipes by setting their PSTATUS.FREEZE bit. Any current on-going pipe will complete its transaction, and then all pipes will be inactive. The user should wait at least 1 complete frame before entering the suspend mode to avoid any data loss.

The device can awaken the host by sending an Upstream Resume (Remote Wakeup feature). When the host detects a non-idle state on the USB bus, it sets the INTFLAG.WAKEUP. If the non-idle bus state corresponds to an Upstream Resume (K state), the Upstream Resume Received Interrupt bit in INTFLAG (INTFLAG.UPRSM) is set and the user must generate a Downstream Resume within 1 ms and for at least 20 ms. It is required to first write a one to the Send USB Resume bit in CTRLB (CTRLB.RESUME) to respond to the upstream resume with a downstream resume. Alternatively, the host can resume from a suspend state by sending a Downstream Resume on the


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

USB bus (CTRLB.RESUME set to 1). In both cases, when the downstream resume is completed, the CTRLB.SOFE bit is automatically set and the host enters again the active state.

#### 38.6.3.8. Phase-locked SOFs

To support the Synchronous Endpoints capability, the period of the emitted Start-of-Frame is maintained while the USB connection is not in the active state. This does not apply for the disconnected/connected/reset states. It applies for active/idle/suspend/resume states. The period of Start-of-Frame will be 1ms when the USB connection is in active state and an integer number of milli-seconds across idle/suspend/resume states.

To ensure the Synchronous Endpoints capability, the GCLK_USB clock must be kept running. If the GCLK_USB is interrupted, the period of the emitted Start-of-Frame will be erratic.

#### 38.6.3.9. Management of Control Pipes

A control transaction is composed of three stages:

- SETUP
- Data (IN or OUT)
- Status (IN or OUT)

The user has to change the pipe token according to each stage using the Pipe Token field in PCFG (PCFG.PTOKEN).

For control pipes only, the token is assigned a specific initial data toggle sequence:

- SETUP: Data0
- IN: Data1
- OUT: Data1

#### 38.6.3.10. Management of IN Pipes

IN packets are sent by the USB device controller upon IN request reception from the host. All the received data from the device to the host will be stored in the bank provided the bank is empty. The pipe and its descriptor in RAM must be configured.

The host indicates it is able to receive data from the device by clearing the Bank 0/1 Ready bit in PSTATUS (PSTATUS.BK0/1RDY), which means that the memory for the bank is available for new USB transfer.

The USB will perform IN requests as long as the pipe is not frozen by the user.

The generation of IN requests starts when the pipe is unfrozen (PSTATUS.PFREEZE is set to zero).

When the current bank is full, the Transmit Complete 0/1 bit in PINTFLAG (PINTFLAG.TRCPT0/1) will be set and trigger an interrupt if enabled and the PSTATUS.BK0/1RDY bit will be set.

PINTFLAG.TRCPT0/1 must be cleared by software to acknowledge the interrupt. This is done by writing a one to the PINTFLAG.TRCPT0/1 of the addressed pipe.

The user reads the PCKSIZE.BYTE_COUNT to know how many bytes should be read.

To free the bank the user must read the IN data from the address ADDR in the pipe descriptor and clear the PKSTATUS.BK0/1RDY bit. When the IN pipe is composed of multiple banks, a successful IN transaction will switch to the next bank. Another IN request will be performed by the host as long as the PSTATUS.BK0/1RDY bit for that bank is set. The PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1RDY will be updated accordingly.

The user can follow the current bank looking at Current Bank bit in PSTATUS (PSTATUS.CURBK) and by looking at Data Toggle for IN pipe bit in PSTATUS (PSTATUS.DTGLIN).

When the pipe is configured as single bank (Pipe Bank bit in PCFG (PCFG.BK) is 0), only PINTFLAG.TRCPT0 and PSTATUS.BK0 are used. When the pipe is configured as dual bank (PCFG.BK is 1), both PINTFLAG.TRCPT0/1 and PSTATUS.BK0/1 are used.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 38.6.3.11. Management of OUT Pipes

OUT packets are sent by the host. All the data stored in the bank will be sent to the device provided the bank is filled. The pipe and its descriptor in RAM must be configured.

The host can send data to the device by writing to the data bank 0 in single bank or the data bank 0/1 in dual bank.

The generation of OUT packet starts when the pipe is unfrozen (PSTATUS.PFREEZE is zero).

The user writes the OUT data to the data buffer pointer by ADDR in the pipe descriptor and allows the USB to send the data by writing a one to the PSTATUS.BK0/1RDY. This will also cause a switch to the next bank if the OUT pipe is part of a dual bank configuration.

PINTFLAGn.TRCPT0/1 must be cleared before setting PSTATUS.BK0/1RDY to avoid missing an PINTFLAGn.TRCPT0/1 event.

#### 38.6.3.12. Alternate Pipe

The user has the possibility to run sequentially several logical pipes on the same physical pipe. It allows addressing of any device endpoint of any attached device on the bus.

Before switching pipe, the user should save the pipe context (Pipe registers and descriptor for pipe n).

After switching pipe, the user should restore the pipe context (Pipe registers and descriptor for pipe n) and in particular PCFG, and PSTATUS.

#### 38.6.3.13. Data Flow Error

This error exists only for isochronous and interrupt pipes for both IN and OUT directions. It sets the Transmit Fail bit in PINTFLAG (PINTFLAG.TRFAIL), which triggers an interrupt if the Transmit Fail bit in PINTENCLR/SET(PINTENCLR/SET.TRFAIL) is set. The user must check the Pipe Interrupt Summary register (PINTSMRY) to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the Pipe Bank Status register (STATUS_BK) for each bank. If the Error Flow bit in the STATUS_BK (STATUS_BK.ERRORFLOW) is set then the user is able to determine the origin of the data flow error. As the user knows that the endpoint is an IN or OUT the error flow can be deduced as OUT underflow or as an IN overflow.

An underflow can occur during an OUT stage if the host attempts to send data from an empty bank. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

An overflow can occur during an IN stage if the device tries to send a packet while the bank is full. Typically this occurs when a CPU is not fast enough. The packet data is not written to the bank and is lost. If a new transaction is successful, the relevant bank descriptor STATUS_BK.ERRORFLOW will be cleared.

#### 38.6.3.14. CRC Error

This error exists only for isochronous IN pipes. It sets the PINTFLAG.TRFAIL, which triggers an interrupt if PINTENCLR/SET.TRFAIL is set. The user must check the PINTSMRY to find out the pipe which triggered the interrupt. Then the user must check the origin of the interrupt's bank by looking at the bank descriptor STATUS_BK for each bank and if the CRC Error bit in STATUS_BK (STATUS_BK.CRCERR) is set then the user is able to determine the origin of the CRC error. A CRC error can occur during the IN stage if the USB detects a corrupted packet. The IN packet will remain stored in the bank and PINTFLAG.TRCPT0/1 will be set.

#### 38.6.3.15. PERR Error

This error exists for all pipes. It sets the PINTFLAG.PERR Interrupt, which triggers an interrupt if PINTFLAG.PERR is set. The user must check the PINTSMRY register to find out the pipe which can cause an interrupt.

A PERR error occurs if one of the error field in the STATUS_PIPE register in the Host pipe descriptor is set and the Error Count field in STATUS_PIPE (STATUS_PIPE.ERCNT) exceeds the maximum allowed


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

number of Pipe errors as defined in Pipe Error Max Number field in CTRL_PIPE (CTRL_PIPE.PERMAX). Refer to the STATUS_PIPE register.

If one of the error field in the STATUS_PIPE register from the Host Pipe Descriptor is set and the STATUS_PIPE.ERCNT is less than the CTRL_PIPE.PERMAX, the STATUS_PIPE.ERCNT is incremented.

#### 38.6.3.16. Link Power Management L1 (LPM-L1) Suspend State Entry and Exit as Host.

An EXTENDED LPM transaction can be transmitted by any enabled pipe. The PCFGn.PTYPE should be set to EXTENDED. Other fields as PCFG.PTOKEN, PCFG.BK and PCKSIZE.SIZE are irrelevant in this configuration. The user should also set the EXTREG.VARIABLE in the descriptor as described in the EXTREG register.

When the pipe is configured and enabled, an EXTENDED TOKEN followed by a LPM TOKEN are transmitted. The device responds with a valid HANDSHAKE, corrupted HANDSHAKE or no HANDSHAKE (TIME-OUT).

If the valid HANDSHAKE is an ACK, the host will immediately proceed to L1 SLEEP and the PINTFLAG.TRCT0 is set. The minimum duration of the L1 SLEEP state will be the TL1RetryAndResidency as defined in the reference document "ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum". When entering the L1 SLEEP state, the CTRLB.SOFE is cleared, avoiding Start-of-Frame generation.

If the valid HANDSHAKE is a NYET PINTFLAG.TRFAIL is set.

If the valid HANDSHAKE is a STALL the PINTFLAG.STALL is set.

If there is no HANDSHAKE or corrupted HANDSHAKE, the EXTENDED/LPM pair of TOKENS will be transmitted again until reaching the maximum number of retries as defined by the CTRL_PIPE.PERMAX in the pipe descriptor.

If the last retry returns no valid HANDSHAKE, the PINTFLAGn.PERR is set, and the STATUS_BK is updated in the pipe descriptor.

All LPM transactions, should they end up with a ACK, a NYET, a STALL or a PERR, will set the PSTATUS.PFREEZE bit, freezing the pipe before a succeeding operation. The user should unfreeze the pipe to start a new LPM transaction.

To exit the L1 STATE, the user initiate a DOWNSTREAM RESUME by setting the bit CTRLB.RESUME or a L1 RESUME by setting the Send L1 Resume bit in CTRLB (CTRLB.L1RESUME). In the case of a L1 RESUME, the K STATE duration is given by the BESL bit field in the EXTREG.VARIABLE field. Refer to the EXTREG register.

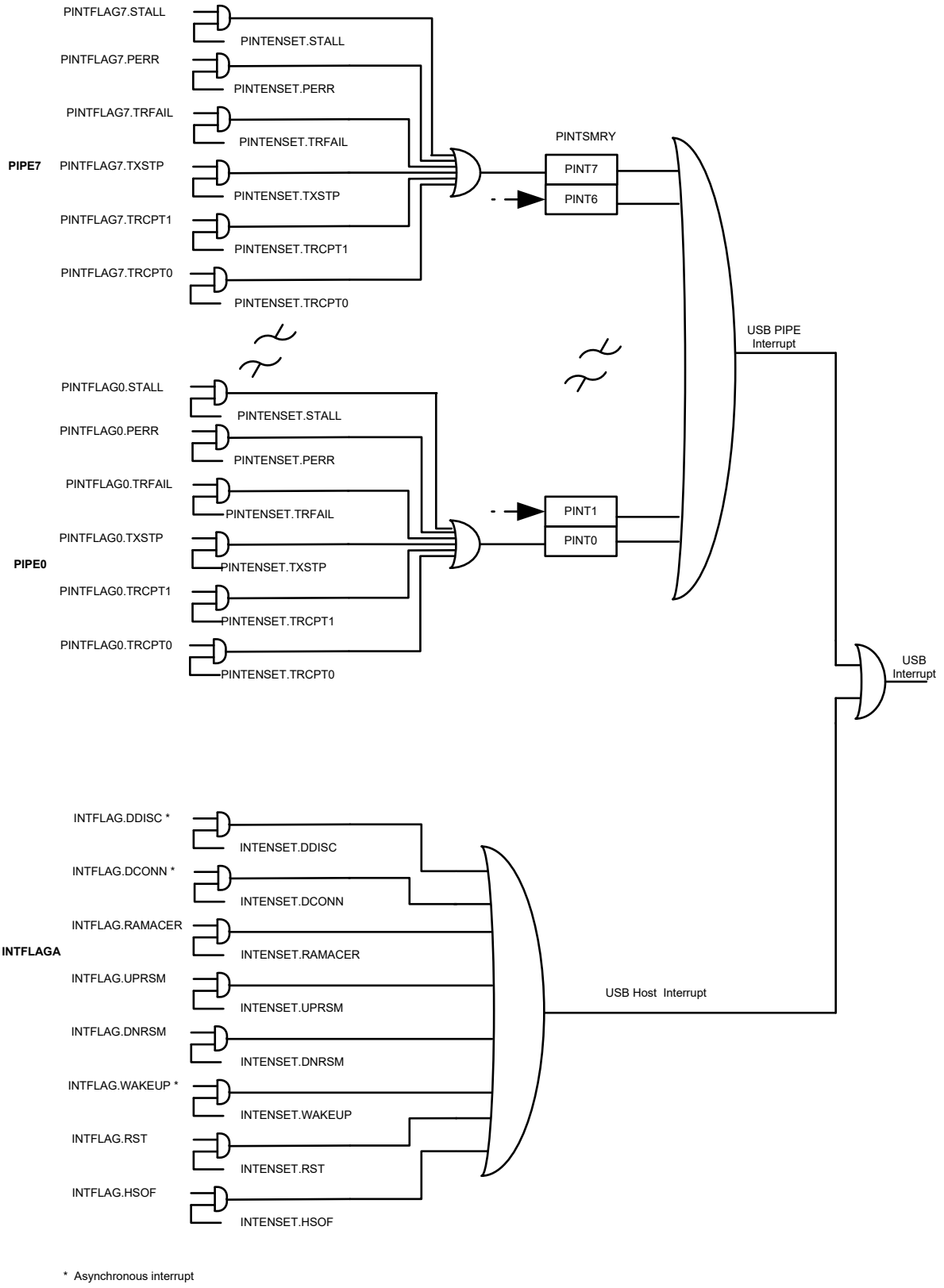
When the host is in the L1 SLEEP state after a successful LPM transmitted, the device can initiate an UPSTREAM RESUME. This will set the Upstream Resume Interrupt bit in INTFLAG (INTFLAG.UPRSM). The host should proceed then to a L1 RESUME as described above.

After resuming from the L1 SLEEP state, the bit CTRLB.SOFE is set, allowing Start-of-Frame generation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.6.3.17. Host Interrupt

**Figure 38-10.** Host Interrupt




The WAKEUP is an asynchronous interrupt and can be used to wake-up the device from any sleep


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.7. Common Device / Host Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | MODE | | | | | RUNSTDBY | ENABLE | SWRST |
| 0x01 | Reserved | | | | | | | | | |
| 0x02 | SYNCBUSY | 7:0 | | | | | | | ENABLE | SWRST |
| 0x03 | QOSCTRL | 7:0 | | | | | DQOS[1:0] | | CQOS[1:0] | |
| 0x04 ... 0x0C | Reserved | | | | | | | | | |
| 0x0D | FSMSTATUS | 7:0 | | FSMSTATE[6:0] | | | | | | |
| 0x0E ... 0x23 | Reserved | | | | | | | | | |
| 0x24 | DESCADD | 7:0 | DESCADD[7:0] | | | | | | | |
| | | 15:8 | DESCADD[15:8] | | | | | | | |
| | | 23:16 | DESCADD[23:16] | | | | | | | |
| | | 31:24 | DESCADD[31:24] | | | | | | | |
| 0x28 | PADCAL | 7:0 | TRANSN[1:0] | | | TRANSP[4:0] | | | | |
| | | 15:8 | | TRIM[2:0] | | | | TRANSN[4:2] | | |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 38.7.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized Bits

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MODE | | | | | RUNSTDBY | ENABLE | SWRST |
| Access | R/W | | | | | R/W | R/W | R/W |
| Reset | 0 | | | | | 0 | 0 | 0 |

**Bit 7 – MODE** Operating Mode
This bit defines the operating mode of the USB.

| Value | Description |
|---|---|
| 0 | USB Device mode |
| 1 | USB Host mode |

**Bit 2 – RUNSTDBY** Run in Standby Mode
This bit is Enable-Protected.

| Value | Description |
|---|---|
| 0 | USB clock is stopped in standby mode. |
| 1 | USB clock is running in standby mode |

**Bit 1 – ENABLE** Enable
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the Synchronization status enable bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is Write-Synchronized.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled or being disabled. |
| 1 | The peripheral is enabled or being enabled. |

**Bit 0 – SWRST** Software Reset
Writing a zero to this bit has no effect.
Writing a '1' to this bit resets all registers in the USB, to their initial state, and the USB will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.
**Notes:**

1. When the CTRLA.SWRST is written, the user must poll the SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without the SWRST are disallowed until the SYNCBUSY.SWRST is cleared by hardware.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.2. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x02
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ENABLE** Synchronization Enable status bit
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.
This bit is set when the synchronization of ENABLE register between clock domains is started.

**Bit 0 – SWRST** Synchronization Software Reset status bit
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.
This bit is set when the synchronization of SWRST register between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.3. QOS Control

**Name:** QOSCTRL
**Offset:** 0x03
**Reset:** 0x000x0F
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DQOS[1:0] | | CQOS[1:0] | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 3:2 – DQOS[1:0]** Data Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write data operation. Refer to *SRAM Quality of Service*.

**Bits 1:0 – CQOS[1:0]** Configuration Quality of Service
These bits define the memory priority access during the endpoint or pipe read/write configuration operation. Refer to *SRAM Quality of Service*.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.4. Finite State Machine Status

**Name:** FSMSTATUS
**Offset:** 0x0D
**Reset:** 0xXXXX
**Property:** Read only

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | FSMSTATE[6:0] | | | | | | |
| Access | | R | R | R | R | R | R | R |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 6:0 – FSMSTATE[6:0]** Fine State Machine Status
These bits indicate the state of the finite state machine of the USB controller.

| Value | Name | Description |
|---|---|---|
| 0x01 | OFF (L3) | Corresponds to the powered-off, disconnected, and disabled state. |
| 0x02 | ON (L0) | Corresponds to the Idle and Active states. |
| 0x04 | SUSPEND (L2) | - |
| 0x08 | SLEEP (L1) | - |
| 0x10 | DNRESUME | Down Stream Resume. |
| 0x20 | UPRESUME | Up Stream Resume. |
| 0x40 | RESET | USB lines Reset. |
| Others | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.5. Descriptor Address

**Name:** DESCADD
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DESCADD[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DESCADD[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DESCADD[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DESCADD[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DESCADD[31:0]** Descriptor Address Value
These bits define the base address of the main USB descriptor in RAM. The two least significant bits must be written to zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.6. Pad Calibration

**Name:** PADCAL
**Offset:** 0x28
**Reset:** 0x0000
**Property:** PAC Write-Protection

The Pad Calibration values must be loaded from the NVM Software Calibration Area into the USB Pad Calibration register by software, before enabling the USB, to achieve the specified accuracy. Refer to NVM Software Calibration Area Mapping (SW0) for additional information.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | TRIM[2:0] | | | | TRANSN[4:2] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TRANSN[1:0] | | | TRANSP[4:0] | | | | |
| Access | R/W | R/W | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 0 |

**Bits 14:12 – TRIM[2:0]** Trim bits for DP/DM
These bits calibrate the matching of rise/fall of DP/DM.

**Bits 10:6 – TRANSN[4:0]** Trimmable Output Driver Impedance N
These bits calibrate the NMOS output impedance of DP/DM drivers.

**Bits 4:0 – TRANSP[4:0]** Trimmable Output Driver Impedance P
These bits calibrate the PMOS output impedance of DP/DM drivers.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.8. General Device Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | CTRLB | 7:0 | | | | NREPLY | SPDCONF[1:0] | | UPRSM | DETACH |
| | | 15:8 | | | | | LPMHDSK[1:0] | | GNAK | |
| 0x0A | DADD | 7:0 | ADDEN | DADD[6:0] | | | | | | |
| 0x0B | Reserved | | | | | | | | | |
| 0x0C | STATUS | 7:0 | LINESTATE[1:0] | | | | SPEED[1:0] | | | |
| 0x0D ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | FNUM | 7:0 | FNUM[4:0] | | | | | | | |
| | | 15:8 | FNCERR | | FNUM[10:5] | | | | | |
| 0x12 ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | RAMACER | UPRSM | EORSM | WAKEUP | EORST | SOF | | SUSPEND |
| | | 15:8 | | | | | | | LPMSUSP | LPMNYET |
| 0x16 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTENSET | 7:0 | RAMACER | UPRSM | EORSM | WAKEUP | EORST | SOF | | SUSPEND |
| | | 15:8 | | | | | | | LPMSUSP | LPMNYET |
| 0x1A ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | INTFLAG | 7:0 | RAMACER | UPRSM | EORSM | WAKEUP | EORST | SOF | | SUSPEND |
| | | 15:8 | | | | | | | LPMSUSP | LPMNYET |
| 0x1E ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | EPINTSMRY | 7:0 | EPINT7 | EPINT6 | EPINT5 | EPINT4 | EPINT3 | EPINT2 | EPINT1 | EPINT0 |
| | | 15:8 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.8.1. Control B

**Name:** CTRLB
**Offset:** 0x08
**Reset:** 0x0001
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LPMHDSK[1:0] | | GNAK | |
| Access | | | | | R/W | R/W | R/W | |
| Reset | | | | | 0 | 0 | 0 | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | NREPLY | SPDCONF[1:0] | | UPRSM | DETACH |
| Access | | | | R | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 1 |

**Bits 11:10 – LPMHDSK[1:0]** Link Power Management Handshake
These bits select the Link Power Management Handshake configuration.

| Value | Description |
|---|---|
| 0x0 | No handshake. LPM is not supported. |
| 0x1 | ACK |
| 0x2 | NYET |
| 0x3 | Reserved |

**Bit 9 – GNAK** Global NAK
This bit configures the operating mode of the NAK.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The handshake packet reports the status of the USB transaction |
| 1 | A NAK handshake is answered for each USB transaction regardless of the current endpoint memory bank status |

**Bit 4 – NREPLY** No reply excepted SETUP Token
This bit is cleared by hardware when receiving a SETUP packet.
This bit has no effect for any other endpoint but endpoint 0.

| Value | Description |
|---|---|
| 0 | Disable the “NO_REPLY” feature: Any transaction to endpoint 0 will be handled according to the USB2.0 standard. |
| 1 | Enable the “NO_REPLY” feature: Any transaction to endpoint 0 will be ignored except SETUP. |

**Bits 3:2 – SPDCONF[1:0]** Speed Configuration
These bits select the speed configuration.

| Value | Description |
|---|---|
| 0x0 | FS: Full-speed |
| 0x1 | LS: Low-speed |
| 0x2 | Reserved |
| 0x3 | Reserved |

**Bit 1 – UPRSM** Upstream Resume
This bit is cleared when the USB receives a USB reset or once the upstream resume has been sent.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Writing a zero to this bit has no effect. |
| 1 | Writing a one to this bit will generate an upstream resume to the host for a remote wakeup. |

**Bit 0 – DETACH** Detach

| Value | Description |
|---|---|
| 0 | The device is attached to the USB bus so that communications may occur. |
| 1 | It is the default value at reset. The internal device pull-ups are disabled, removing the device from the USB bus. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.8.2. Device Address

**Name:** DADD
**Offset:** 0x0A
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDEN | DADD[6:0] | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ADDEN** Device Address Enable
This bit is cleared when a USB reset is received.

| Value | Description |
|---|---|
| 0 | Writing a zero will deactivate the DADD field (USB device address) and return the device to default address 0. |
| 1 | Writing a one will activate the DADD field (USB device address). |

**Bits 6:0 – DADD[6:0]** Device Address
These bits define the device address. The DADD register is reset when a USB reset is received.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.8.3. Status

**Name:** STATUS
**Offset:** 0x0C
**Reset:** 0x40
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LINESTATE[1:0] | | | | SPEED[1:0] | | | |
| Access | R | R | | | R/W | R/W | | |
| Reset | 0 | 1 | | | 0 | 1 | | |

**Bits 7:6 – LINESTATE[1:0]** USB Line State Status
These bits define the current line state DP/DM.

| LINESTATE[1:0] | USB Line Status |
|---|---|
| 0x0 | SE0/RESET |
| 0x1 | FS-J or LS-K State |
| 0x2 | FS-K or LS-J State |

**Bits 3:2 – SPEED[1:0]** Speed Status
These bits define the current speed used of the device.

| SPEED[1:0] | SPEED STATUS |
|---|---|
| 0x0 | Full-speed mode |
| 0x1 | Low-speed mode |
| 0x2 | Reserved |
| 0x3 | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.8.4. Device Frame Number

**Name:** FNUM
**Offset:** 0x10
**Reset:** 0x0000
**Property:** Read only

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FNCERR | | FNUM[10:5] | | | | | |
| Access | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FNUM[4:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | | | |

**Bit 15 – FNCERR** Frame Number CRC Error
This bit is cleared upon receiving a USB reset.
This bit is set when a corrupted frame number (or micro-frame number) is received.
This bit and the SOF (or MSOF) interrupt bit are updated at the same time.

**Bits 13:3 – FNUM[10:0]** Frame Number
These bits are cleared upon receiving a USB reset.
These bits are updated with the frame number information as provided from the last SOF packet even if a corrupted SOF is received.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 38.8.5. Device Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | LPMSUSP | LPMNYET |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RAMACER | UPRSM | EORSM | WAKEUP | EORST | SOF | | SUSPEND |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |

**Bit 9 – LPMSUSP** Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Link Power Management Suspend Interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Link Power Management Suspend interrupt is disabled. |
| 1 | The Link Power Management Suspend interrupt is enabled and an interrupt request will be generated when the Link Power Management Suspend interrupt Flag is set. |

**Bit 8 – LPMNYET** Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Link Power Management Not Yet interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Link Power Management Not Yet interrupt is disabled. |
| 1 | The Link Power Management Not Yet interrupt is enabled and an interrupt request will be generated when the Link Power Management Not Yet interrupt Flag is set. |

**Bit 7 – RAMACER** RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The RAM Access interrupt is disabled. |
| 1 | The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set. |

**Bit 6 – UPRSM** Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Upstream Resume interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set. |

**Bit 5 – EORSM** End Of Resume Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the End Of Resume interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The End Of Resume interrupt is disabled. |
| 1 | The End Of Resume interrupt is enabled and an interrupt request will be generated when the End Of Resume interrupt Flag is set. |

**Bit 4 – WAKEUP** Wake-Up Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Wake Up interrupt is disabled. |
| 1 | The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set. |

**Bit 3 – EORST** End of Reset Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the End of Reset interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The End of Reset interrupt is disabled. |
| 1 | The End of Reset interrupt is enabled and an interrupt request will be generated when the End of Reset interrupt Flag is set. |

**Bit 2 – SOF** Start-of-Frame Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Start-of-Frame interrupt is disabled. |
| 1 | The Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Start-of-Frame interrupt Flag is set. |

**Bit 0 – SUSPEND** Suspend Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Suspend Interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Suspend interrupt is disabled. |
| 1 | The Suspend interrupt is enabled and an interrupt request will be generated when the Suspend interrupt Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.8.6. Device Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x18
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | LPMSUSP | LPMNYET |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RAMACER | UPRSM | EORSM | WAKEUP | EORST | SOF | | SUSPEND |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |

**Bit 9 – LPMSUSP** Link Power Management Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Link Power Management Suspend Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Link Power Management Suspend interrupt is disabled. |
| 1 | The Link Power Management Suspend interrupt is enabled. |

**Bit 8 – LPMNYET** Link Power Management Not Yet Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Link Power Management Not Yet interrupt bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Link Power Management Not Yet interrupt is disabled. |
| 1 | The Link Power Management Not Yet interrupt is enabled. |

**Bit 7 – RAMACER** RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the RAM Access Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The RAM Access interrupt is disabled. |
| 1 | The RAM Access interrupt is enabled. |

**Bit 6 – UPRSM** Upstream Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Upstream Resume Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Upstream Resume interrupt is disabled. |
| 1 | The Upstream Resume interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 5 – EORSM** End Of Resume Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the End Of Resume interrupt Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The End Of Resume interrupt is disabled. |
| 1 | The End Of Resume interrupt is enabled. |

**Bit 4 – WAKEUP** Wake-Up Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Wake Up interrupt is disabled. |
| 1 | The Wake Up interrupt is enabled. |

**Bit 3 – EORST** End of Reset Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the End of Reset interrupt Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The End of Reset interrupt is disabled. |
| 1 | The End of Reset interrupt is enabled. |

**Bit 2 – SOF** Start-of-Frame Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Start-of-Frame interrupt Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Start-of-Frame interrupt is disabled. |
| 1 | The Start-of-Frame interrupt is enabled. |

**Bit 0 – SUSPEND** Suspend Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Suspend interrupt Enable bit and enable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Suspend interrupt is disabled. |
| 1 | The Suspend interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.8.7. Device Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x01C
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | LPMSUSP | LPMNYET |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RAMACER | UPRSM | EORSM | WAKEUP | EORST | SOF | | SUSPEND |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |

**Bit 9 – LPMSUSP** Link Power Management Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB module acknowledge a Link Power Management Transaction (ACK handshake) and has entered the Suspended state and will generate an interrupt if INTENCLR/SET.LPMSUSP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the LPMSUSP Interrupt Flag.

**Bit 8 – LPMNYET** Link Power Management Not Yet Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB module acknowledges a Link Power Management Transaction (handshake is NYET) and will generate an interrupt if INTENCLR/SET.LPMNYET is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the LPMNYET Interrupt Flag.

**Bit 7 – RAMACER** RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a RAM access underflow error occurs during IN data stage. This bit will generate an interrupt if INTENCLR/SET.RAMACER is one.
Writing a zero to this bit has no effect.

**Bit 6 – UPRSM** Upstream Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB sends a resume signal called “Upstream Resume” and will generate an interrupt if INTENCLR/SET.UPRSM is one.
Writing a zero to this bit has no effect.

**Bit 5 – EORSM** End Of Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB detects a valid “End of Resume” signal initiated by the host and will generate an interrupt if INTENCLR/SET.EORSM is one.
Writing a zero to this bit has no effect.

**Bit 4 – WAKEUP** Wake Up Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB is reactivated by a filtered non-idle signal from the lines and will generate an interrupt if INTENCLR/SET.WAKEUP is one.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a zero to this bit has no effect.

**Bit 3 – EORST** End of Reset Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB "End of Reset" has been detected and will generate an interrupt if INTENCLR/SET.EORST is one.
Writing a zero to this bit has no effect.

**Bit 2 – SOF** Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB "Start-of-Frame" has been detected (every 1 ms) and will generate an interrupt if INTENCLR/SET.SOF is one.
The FNUM is updated.
Writing a zero to this bit has no effect.

**Bit 0 – SUSPEND** Suspend Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB "Suspend" idle state has been detected for 3 frame periods (J state for 3 ms) and will generate an interrupt if INTENCLR/SET.SUSPEND is one.
Writing a zero to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.8.8. Endpoint Interrupt Summary

**Name:** EPINTSMRY
**Offset:** 0x20
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EPINT7 | EPINT6 | EPINT5 | EPINT4 | EPINT3 | EPINT2 | EPINT1 | EPINT0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – EPINT** EndPoint Interrupt
The flag EPINT[n] is set when an interrupt is triggered by the EndPoint n. Refer to the EPINTFLAGn register in the Device EndPoint section.
This bit will be cleared when no interrupts are pending for EndPoint n.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.9. Device Endpoint n Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0xFF | Reserved | | | | | | | | | |
| 0x0100 | EPCFG0 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0101 ... 0x0103 | Reserved | | | | | | | | | |
| 0x0104 | EPSTATUSCLR0 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0105 | EPSTATUSSET0 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0106 | EPSTATUS0 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0107 | EPINTFLAG0 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0108 | EPINTENCLR0 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0109 | EPINTENSET0 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x010A ... 0x011F | Reserved | | | | | | | | | |
| 0x0120 | EPCFG1 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0121 ... 0x0123 | Reserved | | | | | | | | | |
| 0x0124 | EPSTATUSCLR1 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0125 | EPSTATUSSET1 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0126 | EPSTATUS1 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0127 | EPINTFLAG1 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0128 | EPINTENCLR1 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0129 | EPINTENSET1 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x012A ... 0x013F | Reserved | | | | | | | | | |
| 0x0140 | EPCFG2 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0141 ... 0x0143 | Reserved | | | | | | | | | |
| 0x0144 | EPSTATUSCLR2 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0145 | EPSTATUSSET2 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0146 | EPSTATUS2 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0147 | EPINTFLAG2 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0148 | EPINTENCLR2 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0149 | EPINTENSET2 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x014A ... 0x015F | Reserved | | | | | | | | | |
| 0x0160 | EPCFG3 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0161 ... 0x0163 | Reserved | | | | | | | | | |
| 0x0164 | EPSTATUSCLR3 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0165 | EPSTATUSSET3 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0166 | EPSTATUS3 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0167 | EPINTFLAG3 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0168 | EPINTENCLR3 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0169 | EPINTENSET3 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x016A ... 0x017F | Reserved | | | | | | | | | |
| 0x0180 | EPCFG4 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0181 ... 0x0183 | Reserved | | | | | | | | | |
| 0x0184 | EPSTATUSCLR4 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Device Endpoint n Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0185 | EPSTATUSSET4 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0186 | EPSTATUS4 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0187 | EPINTFLAG4 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0188 | EPINTENCLR4 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x0189 | EPINTENSET4 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x018A ... 0x019F | Reserved | | | | | | | | | |
| 0x01A0 | EPCFG5 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x01A1 ... 0x01A3 | Reserved | | | | | | | | | |
| 0x01A4 | EPSTATUSCLR5 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01A5 | EPSTATUSSET5 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01A6 | EPSTATUS5 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01A7 | EPINTFLAG5 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01A8 | EPINTENCLR5 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01A9 | EPINTENSET5 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01AA ... 0x01BF | Reserved | | | | | | | | | |
| 0x01C0 | EPCFG6 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x01C1 ... 0x01C3 | Reserved | | | | | | | | | |
| 0x01C4 | EPSTATUSCLR6 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01C5 | EPSTATUSSET6 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01C6 | EPSTATUS6 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01C7 | EPINTFLAG6 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01C8 | EPINTENCLR6 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01C9 | EPINTENSET6 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01CA ... 0x01DF | Reserved | | | | | | | | | |
| 0x01E0 | EPCFG7 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x01E1 ... 0x01E3 | Reserved | | | | | | | | | |
| 0x01E4 | EPSTATUSCLR7 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01E5 | EPSTATUSSET7 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01E6 | EPSTATUS7 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x01E7 | EPINTFLAG7 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01E8 | EPINTENCLR7 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| 0x01E9 | EPINTENSET7 | 7:0 | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.9.1. Device Endpoint Configuration register n

**Name:** EPCFGn
**Offset:** 0x0100 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 6:4 – EPTYPE1[2:0]** Endpoint Type for IN direction
These bits contains the endpoint type for IN direction.
Upon receiving a USB reset EPCFGn.EPTYPE1 is cleared except for endpoint 0 which is unchanged.

| Value | Description |
|---|---|
| 0x0 | Bank1 is disabled. |
| 0x1 | Bank1 is enabled and configured as Control IN. |
| 0x2 | Bank1 is enabled and configured as Isochronous IN. |
| 0x3 | Bank1 is enabled and configured as Bulk IN. |
| 0x4 | Bank1 is enabled and configured as Interrupt IN. |
| 0x5 | Bank1 is enabled and configured as Dual-Bank OUT<br>(Endpoint type is the same as the one defined in EPTYPE0) |
| 0x6-0x7 | Reserved |

**Bits 2:0 – EPTYPE0[2:0]** Endpoint Type for OUT direction
These bits contains the endpoint type for OUT direction.
Upon receiving a USB reset EPCFGn.EPTYPE0 is cleared except for endpoint 0 which is unchanged.

| Value | Description |
|---|---|
| 0x0 | Bank0 is disabled. |
| 0x1 | Bank0 is enabled and configured as Control SETUP / Control OUT. |
| 0x2 | Bank0 is enabled and configured as Isochronous OUT. |
| 0x3 | Bank0 is enabled and configured as Bulk OUT. |
| 0x4 | Bank0 is enabled and configured as Interrupt OUT. |
| 0x5 | Bank0 is enabled and configured as Dual Bank IN<br>(Endpoint type is the same as the one defined in EPTYPE1) |
| 0x6-0x7 | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.9.2. EndPoint Status Clear n

**Name:** EPSTATUSCLRn
**Offset:** 0x0104 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| Access | W | W | W | W | | W | W | W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bit 7 – BK1RDY** Bank 1 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.BK1RDY bit.

**Bit 6 – BK0RDY** Bank 0 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.BK0RDY bit.

**Bit 5 – STALLRQ1** STALL bank 1 Request Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.STALLRQ1 bit.

**Bit 4 – STALLRQ0** STALL bank 0 Request Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.STALLRQ0 bit.

**Bit 2 – CURBK** Current Bank Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.CURBK bit.

**Bit 1 – DTGLIN** Data Toggle IN Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear EPSTATUS.DTGLIN bit.

**Bit 0 – DTGLOUT** Data Toggle OUT Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the EPSTATUS.DTGLOUT bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.9.3. EndPoint Status Set n

**Name:** EPSTATUSSETn
**Offset:** 0x0105 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| Access | W | W | W | W | | W | W | W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bit 7 – BK1RDY** Bank 1 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.BK1RDY bit.

**Bit 6 – BK0RDY** Bank 0 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.BK0RDY bit.

**Bit 5 – STALLRQ1** STALL Request bank 1 Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.STALLRQ1 bit.

**Bit 4 – STALLRQ0** STALL Request bank 0 Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.STALLRQ0 bit.

**Bit 2 – CURBK** Current Bank Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.CURBK bit.

**Bit 1 – DTGLIN** Data Toggle IN Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set EPSTATUS.DTGLIN bit.

**Bit 0 – DTGLOUT** Data Toggle OUT Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the EPSTATUS.DTGLOUT bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.9.4. EndPoint Status n

**Name:** EPSTATUSn
**Offset:** 0x0106 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| Access | R | R | R | R | | R | R | R |
| Reset | 0 | 0 | 0 | 2 | | 0 | 0 | 0 |

**Bit 7 – BK1RDY** Bank 1 is ready
For Control/OUT direction Endpoints, the bank is empty.
Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.

| Value | Description |
|---|---|
| 0 | The bank number 1 is not ready : For IN direction Endpoints, the bank is not yet filled in. |
| 1 | The bank number 1 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full. |

**Bit 6 – BK0RDY** Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.

| Value | Description |
|---|---|
| 0 | The bank number 0 is not ready : For IN direction Endpoints, the bank is not yet filled in. For Control/OUT direction Endpoints, the bank is empty. |
| 1 | The bank number 0 is ready: For IN direction Endpoints, the bank is filled in. For Control/OUT direction Endpoints, the bank is full. |

**Bits 4, 5 – STALLRQ** STALL bank x request
Writing a zero to the bit EPSTATUSCLR.STALLRQ will clear this bit.
Writing a one to the bit EPSTATUSSET.STALLRQ will set this bit.
This bit is cleared by hardware when receiving a SETUP packet.

| Value | Description |
|---|---|
| 0 | Disable STALLRQx feature. |
| 1 | Enable STALLRQx feature: a STALL handshake will be sent to the host in regards to bank x. |

**Bit 2 – CURBK** Current Bank
Writing a zero to the bit EPSTATUSCLR.CURBK will clear this bit.
Writing a one to the bit EPSTATUSSET.CURBK will set this bit.

| Value | Description |
|---|---|
| 0 | The bank0 is the bank that will be used in the next single/multi USB packet. |
| 1 | The bank1 is the bank that will be used in the next single/multi USB packet. |

**Bit 1 – DTGLIN** Data Toggle IN Sequence
Writing a zero to the bit EPSTATUSCLR.DTGLINCLR will clear this bit.
Writing a one to the bit EPSTATUSSET.DTGLINSET will set this bit.

| Value | Description |
|---|---|
| 0 | The PID of the next expected IN transaction will be zero: data 0. |
| 1 | The PID of the next expected IN transaction will be one: data 1. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – DTGLOUT** Data Toggle OUT Sequence

Writing a zero to the bit EPSTATUSCLR.DTGLOUTCLR will clear this bit.
Writing a one to the bit EPSTATUSSET.DTGLOUTSET will set this bit.

| Value | Description |
|---|---|
| 0 | The PID of the next expected OUT transaction will be zero: data 0. |
| 1 | The PID of the next expected OUR transaction will be one: data 1. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.9.5. Device EndPoint Interrupt Flag n

**Name:** EPINTFLAGn
**Offset:** 0x0107 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 2 | 0 | 0 | 2 | 0 | 2 |

**Bits 5, 6 – STALL** Transmit Stall x Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transmit Stall occurs and will generate an interrupt if EPINTENCLR/SET.STALL is one.
EPINTFLAG.STALL is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".
Writing a zero to this bit has no effect.
Writing a one to this bit clears the STALL Interrupt Flag.

**Bit 4 – RXSTP** Received Setup Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Received Setup occurs and will generate an interrupt if EPINTENCLR/SET.RXSTP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the RXSTP Interrupt Flag.

**Bits 2, 3 – TRFAIL** Transfer Fail x Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a transfer fail occurs and will generate an interrupt if EPINTENCLR/SET.TRFAIL is one.
EPINTFLAG.TRFAIL is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRFAIL Interrupt Flag.

**Bits 0, 1 – TRCPT** Transfer Complete x interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer complete occurs and will generate an interrupt if EPINTENCLR/SET.TRCPT is one. EPINTFLAG.TRCPT is set for a single bank OUT endpoint or double bank IN/OUT endpoint when current bank is "0".
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRCPT0 Interrupt Flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.9.6. Device EndPoint Interrupt Enable n

**Name:** EPINTENCLRn
**Offset:** 0x0108 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Set (EPINTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 2 | 0 | 0 | 2 | 0 | 2 |

**Bits 5, 6 – STALL** Transmit STALL x Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transmit Stall x Interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Transmit Stall x interrupt is disabled. |
| 1 | The Transmit Stall x interrupt is enabled and an interrupt request will be generated when the Transmit Stall x Interrupt Flag is set. |

**Bit 4 – RXSTP** Received Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Received Setup Interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Received Setup interrupt is disabled. |
| 1 | The Received Setup interrupt is enabled and an interrupt request will be generated when the Received Setup Interrupt Flag is set. |

**Bits 2, 3 – TRFAIL** Transfer Fail x Interrupt Enable
The user should look into the descriptor table status located in ram to be informed about the error condition : ERRORFLOW, CRC.
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Fail x Interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Transfer Fail bank x interrupt is disabled. |
| 1 | The Transfer Fail bank x interrupt is enabled and an interrupt request will be generated when the Transfer Fail x Interrupt Flag is set. |

**Bits 0, 1 – TRCPT** Transfer Complete x interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete x interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Transfer Complete bank x interrupt is disabled. |
| 1 | The Transfer Complete bank x interrupt is enabled and an interrupt request will be generated when the Transfer Complete x Interrupt Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.9.7. Device Interrupt EndPoint Set n

**Name:** EPINTENSETn
**Offset:** 0x0109 + n*0x20 [n=0..7]
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Endpoint Interrupt Enable Clear (EPINTENCLR) register. This register is cleared by USB reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | STALL1 | STALL0 | RXSTP | TRFAIL1 | TRFAIL0 | TRCPT1 | TRCPT0 |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 2 | 0 | 0 | 2 | 0 | 2 |

**Bits 5, 6 – STALL** Transmit Stall x Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transmit bank x Stall interrupt.

| Value | Description |
|---|---|
| 0 | The Transmit Stall x interrupt is disabled. |
| 1 | The Transmit Stall x interrupt is enabled. |

**Bit 4 – RXSTP** Received Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Received Setup interrupt.

| Value | Description |
|---|---|
| 0 | The Received Setup interrupt is disabled. |
| 1 | The Received Setup interrupt is enabled. |

**Bits 2, 3 – TRFAIL** Transfer Fail bank x Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

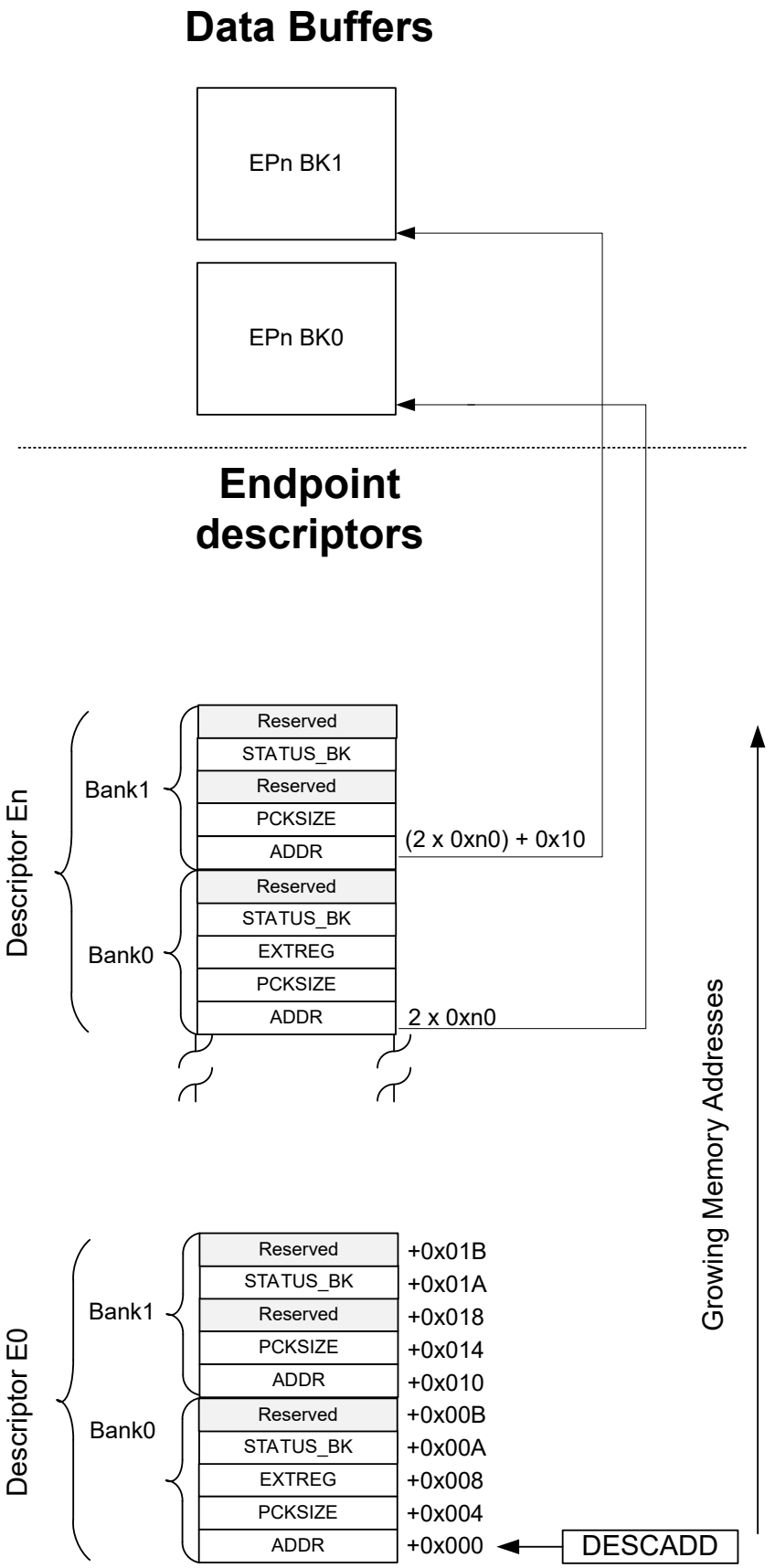
| Value | Description |
|---|---|
| 0 | The Transfer Fail interrupt is disabled. |
| 1 | The Transfer Fail interrupt is enabled. |

**Bits 0, 1 – TRCPT** Transfer Complete bank x interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete x interrupt.
0.2.4 Device Registers - Endpoint RAM

| Value | Description |
|---|---|
| 0 | The Transfer Complete bank x interrupt is disabled. |
| 1 | The Transfer Complete bank x interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.10. Endpoint Descriptors Structure





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.11. Device Endpoint n Descriptor Bank 0/1

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x04 | PCKSIZE | 7:0 | BYTE_COUNT[7:0] | | | | | | | |
| | | 15:8 | MULTI_PACKET_SIZE[1:0] | | | BYTE_COUNT[13:8] | | | | |
| | | 23:16 | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| | | 31:24 | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| 0x08 | EXTREG | 7:0 | VARIABLE[3:0] | | | | SUBPID[3:0] | | | |
| | | 15:8 | | VARIABLE[10:4] | | | | | | |
| 0x0A | STATUS_BK | 7:0 | | | | | | | ERRORFLOW | CRCERR |
| 0x0B ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x14 | PCKSIZE | 7:0 | BYTE_COUNT[7:0] | | | | | | | |
| | | 15:8 | MULTI_PACKET_SIZE[1:0] | | | BYTE_COUNT[13:8] | | | | |
| | | 23:16 | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| | | 31:24 | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| 0x18 ... 0x19 | Reserved | | | | | | | | | |
| 0x1A | STATUS_BK | 7:0 | | | | | | | ERRORFLOW | CRCERR |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.11.1. Address of Data Buffer

**Name:** ADDR
**Offset:** 0x00 + n*0x04 [n=0..1]
**Reset:** 0xXXXXXXXX
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

**Bits 31:0 – ADDR[31:0]** Data Pointer Address Value
These bits define the data pointer address as an absolute word address in RAM. The two least significant bits must be zero to ensure the start address is 32-bit aligned.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.11.2. Packet Size

**Name:** PCKSIZE
**Offset:** 0x04 + n*0x04 [n=0..1]
**Reset:** 0xXXXXXXXX
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | MULTI_PACKET_SIZE[1:0] | | BYTE_COUNT[13:8] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | x | x | x | x | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | BYTE_COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – AUTO_ZLP** Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the endpoint.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for IN endpoints only. When disabled the handshake should be managed by firmware.

| Value | Description |
|---|---|
| 0 | Automatic Zero Length Packet is disabled. |
| 1 | Automatic Zero Length Packet is enabled. |

**Bits 30:28 – SIZE[2:0]** Endpoint size
These bits contains the maximum packet size of the endpoint.

| Value | Description |
|---|---|
| 0x0 | 8 Byte |
| 0x1 | 16 Byte |
| 0x2 | 32 Byte |
| 0x3 | 64 Byte |
| 0x4 | 128 Byte(1) |
| 0x5 | 256 Byte(1) |
| 0x6 | 512 Byte(1) |
| 0x7 | 1023 Byte(1) |
| **Note:** 1. For isochronous endpoint only. | |

**Bits 27:14 – MULTI_PACKET_SIZE[13:0]** Multiple Packet Size
These bits define the 14-bit value that is used for multi-packet transfers.
For IN endpoints, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE should be written to zero when setting up a new transfer.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For OUT endpoints, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

**Bits 13:0 – BYTE_COUNT[13:0]** Byte Count

These bits define the 14-bit value that is used for the byte count.
For IN endpoints, BYTE_COUNT holds the number of bytes to be sent in the next IN transaction.
For OUT endpoint or SETUP endpoints, BYTE_COUNT holds the number of bytes received upon the last OUT or SETUP transaction.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.11.3. Extended Register

**Name:** EXTREG
**Offset:** 0x08
**Reset:** 0xXXXXXXX
**Property:** NA

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | VARIABLE[10:4] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | x | x | x | x | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VARIABLE[3:0] | | | | SUBPID[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | x | x | x | x |

**Bits 14:4 – VARIABLE[10:0]** Variable field send with extended token
These bits define the VARIABLE field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.
To support the USB2.0 Link Power Management addition the VARIABLE field should be read as described below.

| VARIABLES | Description |
|---|---|
| VARIABLE[3:0] | bLinkState (1) |
| VARIABLE[7:4] | BESL (2) |
| VARIABLE[8] | bRemoteWake (1) |
| VARIABLE[10:9] | Reserved |

1. For a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum".
2. For a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

**Bits 3:0 – SUBPID[3:0]** SUBPID field send with extended token
These bits define the SUBPID field of a received extended token. These bits are updated when the USB has answered by an handshake token ACK to a LPM transaction. See Section 2.1.1 Protocol Extension Token in the reference document “ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.11.4. Device Status Bank

**Name:** STATUS_BK
**Offset:** 0x0A + n*0x10 [n=0..1]
**Reset:** 0xXXXXXXX
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ERRORFLOW | CRCERR |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | x | x |

**Bit 1 – ERRORFLOW** Error Flow Status

This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For OUT transfer, a NAK handshake has been sent.
For Isochronous OUT transfer, an overrun condition has occurred.
For IN transfer, this bit is not valid. EPSTATUS.TRFAIL0 and EPSTATUS.TRFAIL1 should reflect the flow errors.

| Value | Description |
|---|---|
| 0 | No Error Flow detected. |
| 1 | A Error Flow has been detected. |

**Bit 0 – CRCERR** CRC Error

This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous OUT endpoint bank.
0.2.5 Host Registers - Common

| Value | Description |
|---|---|
| 0 | No CRC Error. |
| 1 | CRC Error detected. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.12. General Host Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | CTRLB | 7:0 | | | | AUTORESUME | SPDCONF[1:0] | | RESUME | |
| | | 15:8 | | | | | L1RESUME | VBUSOK | BUSRESET | SOFE |
| 0x0A | HSOFC | 7:0 | FLENCE | | | | FLENC[3:0] | | | |
| 0x0B | Reserved | | | | | | | | | |
| 0x0C | STATUS | 7:0 | LINESTATE[1:0] | | | | SPEED[1:0] | | | |
| 0x0D ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | FNUM | 7:0 | FNUM[4:0] | | | | | | | |
| | | 15:8 | | | FNUM[10:5] | | | | | |
| 0x12 | FLENHIGH | 7:0 | FLENHIGH[7:0] | | | | | | | |
| 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | RAMACER | UPRSM | DNRSM | WAKEUP | RST | HSOF | | |
| | | 15:8 | | | | | | | DDISC | DCONN |
| 0x16 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTENSET | 7:0 | RAMACER | UPRSM | DNRSM | WAKEUP | RST | HSOF | | |
| | | 15:8 | | | | | | | DDISC | DCONN |
| 0x1A ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | INTFLAG | 7:0 | RAMACER | UPRSM | DNRSM | WAKEUP | RST | HSOF | | |
| | | 15:8 | | | | | | | DDISC | DCONN |
| 0x1E ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | PINTSMRY | 7:0 | EPINT7 | EPINT6 | EPINT5 | EPINT4 | EPINT3 | EPINT2 | EPINT1 | EPINT0 |
| | | 15:8 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.12.1. Control B

**Name:** CTRLB
**Offset:** 0x08
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | L1RESUME | VBUSOK | BUSRESET | SOFE |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | AUTORESUME | SPDCONF[1:0] | | RESUME | |
| Access | | | | R/W | R/W | R/W | R/W | |
| Reset | | | | 0 | 0 | 0 | 0 | |

**Bit 11 – L1RESUME** Send USB L1 Resume
Writing 0 to this bit has no effect.
1: Generates a USB L1 Resume on the USB bus. This bit should only be set when the Start-of-Frame generation is enabled (SOFE bit set). The duration of the USB L1 Resume is defined by the EXTREG.VARIABLE[7:4] bits field also known as BESL (See LPM ECN).See the EXTREG Register.
This bit is cleared when the USB L1 Resume has been sent or when a USB reset is requested.

**Bit 10 – VBUSOK** VBUS is OK
This notifies the USB HOST that USB operations can be started. When this bit is zero and even if the USB HOST is configured and enabled, HOST operation is halted. Setting this bit will allow HOST operation when the USB is configured and enabled.

| Value | Description |
|---|---|
| 0 | The USB module is notified that the VBUS on the USB line is not powered. |
| 1 | The USB module is notified that the VBUS on the USB line is powered. |

**Bit 9 – BUSRESET** Send USB Reset

| Value | Description |
|---|---|
| 0 | Reset generation is disabled. It is written to zero when the USB reset is completed or when a device disconnection is detected. Writing zero has no effect. |
| 1 | Generates a USB Reset on the USB bus. |

**Bit 8 – SOFE** Start-of-Frame Generation Enable

| Value | Description |
|---|---|
| 0 | The SOF generation is disabled and the USB bus is in suspend state. |
| 1 | Generates SOF on the USB bus in full speed and keep it alive in low speed mode. This bit is automatically set at the end of a USB reset (INTFLAG.RST) or at the end of a downstream resume (INTFLAG.DNRSM) or at the end of L1 resume. |

**Bit 4 – AUTORESUME** Auto Resume Enable

| Value | Description |
|---|---|
| 0 | The Auto Resume is disabled. |
| 1 | Enable Auto Resume |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 3:2 – SPDCONF[1:0]** Speed Configuration for Host
These bits select the host speed configuration as shown below

| Value | Description |
|---|---|
| 0x0 | Low and Full Speed capable |
| 0x1 | Reserved |
| 0x2 | Reserved |
| 0x3 | Reserved |

**Bit 1 – RESUME** Send USB Resume
Writing 0 to this bit has no effect.
1: Generates a USB Resume on the USB bus.
This bit is cleared when the USB Resume has been sent or when a USB reset is requested.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 38.12.2. Host Start-of-Frame Control

**Name:** HSOFC
**Offset:** 0x0A
**Reset:** 0x00
**Property:** PAC Write-Protection

During a very short period just before transmitting a Start-of-Frame, this register is locked. Thus, after writing, it is recommended to check the register value, and write this register again if necessary. This register is cleared upon a USB reset.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FLENCE | | | | FLENC[3:0] | | | |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 7 – FLENCE** Frame Length Control Enable
When this bit is '1', the time between Start-of-Frames can be tuned by up to +/-0.06% using FLENC[3:0].
**Note:** In Low Speed mode, FLENCE must be '0'.

| Value | Description |
|---|---|
| 0 | Start-of-Frame is generated every 1ms. |
| 1 | Start-of-Frame generation depends on the signed value of FLENC[3:0].<br>USB Start-of-Frame period equals 1ms + (FLENC[3:0]/12000)ms |

**Bits 3:0 – FLENC[3:0]** Frame Length Control
These bits define the signed value of the 4-bit FLENC that is added to the Internal Frame Length when FLENCE is '1'. The internal Frame length is the top value of the frame counter when FLENCE is zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.12.3. Status

**Name:** STATUS
**Offset:** 0x0C
**Reset:** 0x00
**Property:** Read only

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | LINESTATE[1:0] | | | | SPEED[1:0] | | | |
| Access | R | R | | | R/W | R/W | | |
| Reset | 0 | 0 | | | 0 | 0 | | |

**Bits 7:6 – LINESTATE[1:0]** USB Line State Status
These bits define the current line state DP/DM.

| LINESTATE[1:0] | USB Line Status |
|---|---|
| 0x0 | SE0/RESET |
| 0x1 | FS-J or LS-K State |
| 0x2 | FS-K or LS-J State |

**Bits 3:2 – SPEED[1:0]** Speed Status
These bits define the current speed used by the host.

| SPEED[1:0] | Speed Status |
|---|---|
| 0x0 | Full-speed mode |
| 0x1 | Low-speed mode |
| 0x2 | Reserved |
| 0x3 | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.12.4. Host Frame Number

**Name:** FNUM
**Offset:** 0x10
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | FNUM[10:5] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FNUM[4:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | | | |

**Bits 13:3 – FNUM[10:0]** Frame Number

These bits contains the current SOF number.
These bits can be written by software to initialize a new frame number value. In this case, at the next SOF, the FNUM field takes its new value and the MFNUM bits are cleared.
As the FNUM register lies across two consecutive byte addresses, writing byte-wise (8-bits) to the FNUM register may produce incorrect frame number generation. It is recommended to write FNUM register word-wise (32-bits) or half-word-wise (16-bits).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.12.5. Host Frame Length

**Name:** FLENHIGH
**Offset:** 0x12
**Reset:** 0x00
**Property:** Read-Only

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FLENHIGH[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – FLENHIGH[7:0]** Frame Length
These bits contains the 8 high-order bits of the internal frame counter.

**Table 38-1.** Counter Description vs. Speed

| Host Register STATUS.SPEED | Description |
|---|---|
| Full Speed | With a USB clock running at 12 MHz, counter length is 12000 to ensure a SOF generation every 1 ms. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 38.12.6. Host Interrupt Enable Register Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DDISC | DCONN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RAMACER | UPRSM | DNRSM | WAKEUP | RST | HSOF | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bit 9 – DDISC** Device Disconnection Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Device Disconnection interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Device Disconnection interrupt is disabled. |
| 1 | The Device Disconnection interrupt is enabled and an interrupt request will be generated when the Device Disconnection interrupt Flag is set. |

**Bit 8 – DCONN** Device Connection Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Device Connection interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Device Connection interrupt is disabled. |
| 1 | The Device Connection interrupt is enabled and an interrupt request will be generated when the Device Connection interrupt Flag is set. |

**Bit 7 – RAMACER** RAM Access Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the RAM Access interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The RAM Access interrupt is disabled. |
| 1 | The RAM Access interrupt is enabled and an interrupt request will be generated when the RAM Access interrupt Flag is set. |

**Bit 6 – UPRSM** Upstream Resume from Device Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Upstream Resume interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Upstream Resume interrupt is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The Upstream Resume interrupt is enabled and an interrupt request will be generated when the Upstream Resume interrupt Flag is set. |

**Bit 5 – DNRSM** Down Resume Interrupt Disable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Down Resume interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Down Resume interrupt is disabled. |
| 1 | The Down Resume interrupt is enabled and an interrupt request will be generated when the Down Resume interrupt Flag is set. |

**Bit 4 – WAKEUP** Wake Up Interrupt Disable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Wake Up interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Wake Up interrupt is disabled. |
| 1 | The Wake Up interrupt is enabled and an interrupt request will be generated when the Wake Up interrupt Flag is set. |

**Bit 3 – RST** BUS Reset Interrupt Disable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Bus Reset interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Bus Reset interrupt is disabled. |
| 1 | The Bus Reset interrupt is enabled and an interrupt request will be generated when the Bus Reset interrupt Flag is set. |

**Bit 2 – HSOF** Host Start-of-Frame Interrupt Disable

Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Host Start-of-Frame interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Host Start-of-Frame interrupt is disabled. |
| 1 | The Host Start-of-Frame interrupt is enabled and an interrupt request will be generated when the Host Start-of-Frame interrupt Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.12.7. Host Interrupt Enable Register Set

**Name:** INTENSET
**Offset:** 0x18
**Reset:** 0x0000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DDISC | DCONN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RAMACER | UPRSM | DNRSM | WAKEUP | RST | HSOF | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bit 9 – DDISC** Device Disconnection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Device Disconnection interrupt bit and enable the DDSIC interrupt.

| Value | Description |
|---|---|
| 0 | The Device Disconnection interrupt is disabled. |
| 1 | The Device Disconnection interrupt is enabled. |

**Bit 8 – DCONN** Device Connection Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Device Connection interrupt bit and enable the DCONN interrupt.

| Value | Description |
|---|---|
| 0 | The Device Connection interrupt is disabled. |
| 1 | The Device Connection interrupt is enabled. |

**Bit 7 – RAMACER** RAM Access Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the RAM Access interrupt bit and enable the RAMACER interrupt.

| Value | Description |
|---|---|
| 0 | The RAM Access interrupt is disabled. |
| 1 | The RAM Access interrupt is enabled. |

**Bit 6 – UPRSM** Upstream Resume from the device Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Upstream Resume interrupt bit and enable the UPRSM interrupt.

| Value | Description |
|---|---|
| 0 | The Upstream Resume interrupt is disabled. |
| 1 | The Upstream Resume interrupt is enabled. |

**Bit 5 – DNRSM** Down Resume Interrupt Enable
Writing a zero to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a one to this bit will set the Down Resume interrupt Enable bit and enable the DNRSM interrupt.

| Value | Description |
|---|---|
| 0 | The Down Resume interrupt is disabled. |
| 1 | The Down Resume interrupt is enabled. |

**Bit 4 – WAKEUP** Wake Up Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Wake Up interrupt Enable bit and enable the WAKEUP interrupt request.

| Value | Description |
|---|---|
| 0 | The WakeUp interrupt is disabled. |
| 1 | The WakeUp interrupt is enabled. |

**Bit 3 – RST** Bus Reset Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Bus Reset interrupt Enable bit and enable the Bus RST interrupt.

| Value | Description |
|---|---|
| 0 | The Bus Reset interrupt is disabled. |
| 1 | The Bus Reset interrupt is enabled. |

**Bit 2 – HSOF** Host Start-of-Frame Interrupt Enable

Writing a zero to this bit has no effect.
Writing a one to this bit will set the Host Start-of-Frame interrupt Enable bit and enable the HSOF interrupt.

| Value | Description |
|---|---|
| 0 | The Host Start-of-Frame interrupt is disabled. |
| 1 | The Host Start-of-Frame interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.12.8. Host Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x1C
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DDISC | DCONN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RAMACER | UPRSM | DNRSM | WAKEUP | RST | HSOF | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

**Bit 9 – DDISC** Device Disconnection Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the device has been removed from the USB Bus and will generate an interrupt if INTENCLR/SET.DDISC is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the DDISC Interrupt Flag.

**Bit 8 – DCONN** Device Connection Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a new device has been connected to the USB BUS and will generate an interrupt if INTENCLR/SET.DCONN is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the DCONN Interrupt Flag.

**Bit 7 – RAMACER** RAM Access Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a RAM access error occurs during an OUT stage and will generate an interrupt if INTENCLR/SET.RAMACER is one.
Writing a zero to this bit has no effect.

**Bit 6 – UPRSM** Upstream Resume from the Device Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB has received an Upstream Resume signal from the Device and will generate an interrupt if INTENCLR/SET.UPRSM is one.
Writing a zero to this bit has no effect.

**Bit 5 – DNRSM** Down Resume Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when the USB has sent a Down Resume and will generate an interrupt if INTENCLR/SET.DRSM is one.
Writing a zero to this bit has no effect.

**Bit 4 – WAKEUP** Wake Up Interrupt Flag
This flag is cleared by writing a one.
This flag is set when:
l The host controller is in suspend mode (SOFE is zero) and an upstream resume from the device is detected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

l The host controller is in suspend mode (SOFE is zero) and an device disconnection is detected.
l The host controller is in operational state (VBUSOK is one) and an device connection is detected.
In all cases it will generate an interrupt if INTENCLR/SET.WAKEUP is one.
Writing a zero to this bit has no effect.

**Bit 3 – RST** Bus Reset Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Bus "Reset" has been sent to the Device and will generate an interrupt if INTENCLR/SET.RST is one.
Writing a zero to this bit has no effect.

**Bit 2 – HSOF** Host Start-of-Frame Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a USB "Host Start-of-Frame" in Full Speed or a keep-alive in Low Speed has been sent (every 1 ms) and will generate an interrupt if INTENCLR/SET.HSOF is one.
The value of the FNUM register is updated.
Writing a zero to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.12.9. Pipe Interrupt Summary

**Name:** PINTSMRY
**Offset:** 0x20
**Reset:** 0x0000
**Property:** Read-only

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EPINT7 | EPINT6 | EPINT5 | EPINT4 | EPINT3 | EPINT2 | EPINT1 | EPINT0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – EPINT**

The flag EPINTn is set when an interrupt is triggered by the pipe n. See the PINTFLAG register in the Host Pipe Register section.
This bit will be cleared when there are no interrupts pending for Pipe n.
Writing to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.13. Host Pipe n Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0xFF | Reserved | | | | | | | | | |
| 0x0100 | PCFG0 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0101 ... 0x0102 | Reserved | | | | | | | | | |
| 0x0103 | BINTERVAL0 | 7:0 | | | | BINTERVAL[7:0] | | | | |
| 0x0104 | PSTATUSCLR0 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0105 | PSTATUSSET0 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0106 | PSTATUS0 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0107 | PINTFLAG0 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0108 | PINTENCLR0 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0109 | PINTENSET0 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x010A ... 0x011F | Reserved | | | | | | | | | |
| 0x0120 | PCFG1 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0121 ... 0x0122 | Reserved | | | | | | | | | |
| 0x0123 | BINTERVAL1 | 7:0 | | | | BINTERVAL[7:0] | | | | |
| 0x0124 | PSTATUSCLR1 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0125 | PSTATUSSET1 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0126 | PSTATUS1 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0127 | PINTFLAG1 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0128 | PINTENCLR1 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0129 | PINTENSET1 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x012A ... 0x013F | Reserved | | | | | | | | | |
| 0x0140 | PCFG2 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0141 ... 0x0142 | Reserved | | | | | | | | | |
| 0x0143 | BINTERVAL2 | 7:0 | | | | BINTERVAL[7:0] | | | | |
| 0x0144 | PSTATUSCLR2 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0145 | PSTATUSSET2 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0146 | PSTATUS2 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0147 | PINTFLAG2 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0148 | PINTENCLR2 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0149 | PINTENSET2 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x014A ... 0x015F | Reserved | | | | | | | | | |
| 0x0160 | PCFG3 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0161 ... 0x0162 | Reserved | | | | | | | | | |
| 0x0163 | BINTERVAL3 | 7:0 | | | | BINTERVAL[7:0] | | | | |
| 0x0164 | PSTATUSCLR3 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0165 | PSTATUSSET3 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0166 | PSTATUS3 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0167 | PINTFLAG3 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0168 | PINTENCLR3 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0169 | PINTENSET3 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x016A ... 0x017F | Reserved | | | | | | | | | |
| 0x0180 | PCFG4 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Host Pipe n Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0181 ... 0x0182 | Reserved | | | | | | | | | |
| 0x0183 | BINTERVAL4 | 7:0 | BINTERVAL[7:0] | | | | | | | |
| 0x0184 | PSTATUSCLR4 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0185 | PSTATUSSET4 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0186 | PSTATUS4 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0187 | PINTFLAG4 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0188 | PINTENCLR4 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x0189 | PINTENSET4 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x018A ... 0x019F | Reserved | | | | | | | | | |
| 0x01A0 | PCFG5 | 7:0 | | | PTYPE[2:0] | | | BK | PTOKEN[1:0] | |
| 0x01A1 ... 0x01A2 | Reserved | | | | | | | | | |
| 0x01A3 | BINTERVAL5 | 7:0 | BINTERVAL[7:0] | | | | | | | |
| 0x01A4 | PSTATUSCLR5 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01A5 | PSTATUSSET5 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01A6 | PSTATUS5 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01A7 | PINTFLAG5 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01A8 | PINTENCLR5 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01A9 | PINTENSET5 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01AA ... 0x01BF | Reserved | | | | | | | | | |
| 0x01C0 | PCFG6 | 7:0 | | | PTYPE[2:0] | | | BK | PTOKEN[1:0] | |
| 0x01C1 ... 0x01C2 | Reserved | | | | | | | | | |
| 0x01C3 | BINTERVAL6 | 7:0 | BINTERVAL[7:0] | | | | | | | |
| 0x01C4 | PSTATUSCLR6 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01C5 | PSTATUSSET6 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01C6 | PSTATUS6 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01C7 | PINTFLAG6 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01C8 | PINTENCLR6 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01C9 | PINTENSET6 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01CA ... 0x01DF | Reserved | | | | | | | | | |
| 0x01E0 | PCFG7 | 7:0 | | | PTYPE[2:0] | | | BK | PTOKEN[1:0] | |
| 0x01E1 ... 0x01E2 | Reserved | | | | | | | | | |
| 0x01E3 | BINTERVAL7 | 7:0 | BINTERVAL[7:0] | | | | | | | |
| 0x01E4 | PSTATUSCLR7 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01E5 | PSTATUSSET7 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01E6 | PSTATUS7 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x01E7 | PINTFLAG7 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01E8 | PINTENCLR7 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| 0x01E9 | PINTENSET7 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.13.1. Host Pipe n Configuration

**Name:** PCFGn
**Offset:** 0x0100 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | PTYPE[2:0] | | | BK | PTOKEN[1:0] | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:3 – PTYPE[2:0]** Type of the Pipe
These bits contains the pipe type.

| PTYPE[2:0] | Description |
|---|---|
| 0x0 | Pipe is disabled |
| 0x1 | Pipe is enabled and configured as CONTROL |
| 0x2 | Pipe is enabled and configured as ISO |
| 0x3 | Pipe is enabled and configured as BULK |
| 0x4 | Pipe is enabled and configured as INTERRUPT |
| 0x5 | Pipe is enabled and configured as EXTENDED |
| 0x06-0x7 | Reserved |

These bits are cleared upon sending a USB reset.

**Bit 2 – BK** Pipe Bank
This bit selects the number of banks for the pipe.
For control endpoints writing a zero to this bit is required as only Bank0 is used for Setup/In/Out transactions.
This bit is cleared when a USB reset is sent.

| BK (1) | Description |
|---|---|
| 0x0 | Single-bank endpoint |
| 0x1 | Dual-bank endpoint |

1. Bank field is ignored when PTYPE is configured as EXTENDED.

| Value | Description |
|---|---|
| 0 | A single bank is used for the pipe. |
| 1 | A dual bank is used for the pipe. |

**Bits 1:0 – PTOKEN[1:0]** Pipe Token
These bits contains the pipe token.

| PTOKEN[1:0](1) | Description |
|---|---|
| 0x0 | SETUP(2) |
| 0x1 | IN |
| 0x2 | OUT |
| 0x3 | Reserved |

1. PTOKEN field is ignored when PTYPE is configured as EXTENDED.
2. Available only when PTYPE is configured as CONTROL

Theses bits are cleared upon sending a USB reset.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.13.2. Interval for the Bulk-Out/Ping Transaction for Endpoint n

**Name:** BINTERVALn
**Offset:** 0x0103 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BINTERVAL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – BINTERVAL[7:0]** BINTERVAL
These bits contains the Ping/Bulk-out period.
These bits are cleared when a USB reset is sent or when PEN[n] is zero.

| BINTERVAL | Description |
|---|---|
| = 0 | Multiple consecutive OUT token is sent in the same frame until it is acked by the peripheral |
| > 0 | One OUT token is sent every BINTERVAL frame until it is acked by the peripheral |

Depending on the type of pipe, the desired period is defined as follows:

| PTYPE | Description |
|---|---|
| Interrupt | 1 ms to 255 ms |
| Isochronous | 2^(Binterval) * 1 ms |
| Bulk or control | 1 ms to 255 ms |
| EXT LPM | bInterval ignored. Always 1 ms when a NYET is received. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.13.3. Pipe Status Clear n

**Name:** PSTATUSCLRn
**Offset:** 0x0104 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| Access | W | W | | W | | W | | W |
| Reset | 0 | 0 | | 0 | | 0 | | 0 |

**Bit 7 – BK1RDY** Bank 1 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.BK1RDY bit.

**Bit 6 – BK0RDY** Bank 0 Ready Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.BK0RDY bit.

**Bit 4 – PFREEZE** Pipe Freeze Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.PFREEZE bit.

**Bit 2 – CURBK** Current Bank Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.CURBK bit.

**Bit 0 – DTGL** Data Toggle Clear
Writing a zero to this bit has no effect.
Writing a one to this bit will clear PSTATUS.DTGL bit.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 38.13.4. Pipe Status Set Register n

**Name:** PSTATUSSETn
**Offset:** 0x0105 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| Access | W | W | | W | | W | | W |
| Reset | 0 | 0 | | 0 | | 0 | | 0 |

**Bit 7 – BK1RDY** Bank 1 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the bit PSTATUS.BK1RDY.

**Bit 6 – BK0RDY** Bank 0 Ready Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set the bit PSTATUS.BK0RDY.

**Bit 4 – PFREEZE** Pipe Freeze Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.PFREEZE bit.

**Bit 2 – CURBK** Current Bank Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.CURBK bit.

**Bit 0 – DTGL** Data Toggle Set
Writing a zero to this bit has no effect.
Writing a one to this bit will set PSTATUS.DTGL bit.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 38.13.5. Pipe Status Register n

**Name:** PSTATUSn
**Offset:** 0x0106 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| Access | R | R | | R | | R | | R |
| Reset | 0 | 0 | | 0 | | 0 | | 0 |

**Bit 7 – BK1RDY** Bank 1 is ready
Writing a one to the bit EPSTATUSCLR.BK1RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK1RDY will set this bit.
This bank is not used for Control pipe.

| Value | Description |
|---|---|
| 0 | The bank number 1 is not ready: For IN the bank is empty. For Control/OUT the bank is not yet fill in. |
| 1 | The bank number 1 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in. |

**Bit 6 – BK0RDY** Bank 0 is ready
Writing a one to the bit EPSTATUSCLR.BK0RDY will clear this bit.
Writing a one to the bit EPSTATUSSET.BK0RDY will set this bit.
This bank is the only one used for Control pipe.

| Value | Description |
|---|---|
| 0 | The bank number 0 is not ready: For IN the bank is not empty. For Control/OUT the bank is not yet fill in. |
| 1 | The bank number 0 is ready: For IN the bank is filled full. For Control/OUT the bank is filled in. |

**Bit 4 – PFREEZE** Pipe Freeze
Writing a one to the bit EPSTATUSCLR.PFREEZE will clear this bit.
Writing a one to the bit EPSTATUSSET.PFREEZE will set this bit.
This bit is also set by the hardware:

- When a STALL handshake has been received.
- After a PIPE has been enabled (rising of bit PEN.N).
- When an LPM transaction has completed whatever handshake is returned or the transaction was timed-out.
- When a pipe transfer was completed with a pipe error. See the PINTFLAG register.

When PFREEZE bit is set while a transaction is in progress on the USB bus, this transaction will be properly completed. PFREEZE bit will be read as "1" only when the ongoing transaction will have been completed.

| Value | Description |
|---|---|
| 0 | The Pipe operates in normal operation. |
| 1 | The Pipe is frozen and no additional requests will be sent to the device on this pipe address. |

**Bit 2 – CURBK** Current Bank

| Value | Description |
|---|---|
| 0 | The bank0 is the bank that will be used in the next single/multi USB packet. |
| 1 | The bank1 is the bank that will be used in the next single/multi USB packet. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – DTGL** Data Toggle Sequence

Writing a one to the bit EPSTATUSCLR.DTGL will clear this bit.
Writing a one to the bit EPSTATUSSET.DTGL will set this bit.
This bit is toggled automatically by hardware after a data transaction.
This bit will reflect the data toggle in regards of the token type (IN/OUT/SETUP).

| Value | Description |
|---|---|
| 0 | The PID of the next expected transaction will be zero: data 0. |
| 1 | The PID of the next expected transaction will be one: data 1. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 38.13.6. Host Pipe Interrupt Flag Register

**Name:** PINTFLAGn
**Offset:** 0x0107 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 2 |

**Bit 5 – STALL** STALL Received Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a stall occurs and will generate an interrupt if PINTENCLR/SET.STALL is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the STALL Interrupt Flag.

**Bit 4 – TXSTP** Transmitted Setup Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer Complete occurs and will generate an interrupt if PINTENCLR/SET.TXSTP is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TXSTP Interrupt Flag.

**Bit 3 – PERR** Pipe Error Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a pipe error occurs and will generate an interrupt if PINTENCLR/SET.PERR is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the PERR Interrupt Flag.

**Bit 2 – TRFAIL** Transfer Fail Interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer Fail occurs and will generate an interrupt if PINTENCLR/SET.TRFAIL is one.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRFAIL Interrupt Flag.

**Bits 0, 1 – TRCPT** Transfer Complete x interrupt Flag
This flag is cleared by writing a one to the flag.
This flag is set when a Transfer complete occurs and will generate an interrupt if PINTENCLR/SET.TRCPT is one. PINTFLAG.TRCPT is set for a single bank IN/OUT pipe or a double bank IN/OUT pipe when current bank is 0.
Writing a zero to this bit has no effect.
Writing a one to this bit clears the TRCPT Interrupt Flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.13.7. Host Pipe Interrupt Clear Register

**Name:** PINTENCLRn
**Offset:** 0x0108 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENSET) register.

This register is cleared by USB reset or when PEN[n] is zero.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 2 |

**Bit 5 – STALL** Received Stall Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Received Stall interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The received Stall interrupt is disabled. |
| 1 | The received Stall interrupt is enabled and an interrupt request will be generated when the received Stall interrupt Flag is set. |

**Bit 4 – TXSTP** Transmitted Setup Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transmitted Setup interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Transmitted Setup interrupt is disabled. |
| 1 | The Transmitted Setup interrupt is enabled and an interrupt request will be generated when the Transmitted Setup interrupt Flag is set. |

**Bit 3 – PERR** Pipe Error Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Pipe Error interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Pipe Error interrupt is disabled. |
| 1 | The Pipe Error interrupt is enabled and an interrupt request will be generated when the Pipe Error interrupt Flag is set. |

**Bit 2 – TRFAIL** Transfer Fail Interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Fail interrupt Enable bit and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Transfer Fail interrupt is disabled. |
| 1 | The Transfer Fail interrupt is enabled and an interrupt request will be generated when the Transfer Fail interrupt Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 0, 1 – TRCPT** Transfer Complete Bank x interrupt Disable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Transfer Complete interrupt Enable bit x and disable the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Transfer Complete Bank x interrupt is disabled. |
| 1 | The Transfer Complete Bank x interrupt is enabled and an interrupt request will be generated when the Transfer Complete interrupt x Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.13.8. Host Interrupt Pipe Set Register

**Name:** PINTENSETn
**Offset:** 0x0109 + n*0x20 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Pipe Interrupt Enable Set (PINTENCLR) register.

This register is cleared by USB reset or when PEN[n] is zero.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | STALL | TXSTP | PERR | TRFAIL | TRCPT1 | TRCPT0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 2 |

**Bit 5 – STALL** Stall Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Stall interrupt.

| Value | Description |
|---|---|
| 0 | The Stall interrupt is disabled. |
| 1 | The Stall interrupt is enabled. |

**Bit 4 – TXSTP** Transmitted Setup Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transmitted Setup interrupt.

| Value | Description |
|---|---|
| 0 | The Transmitted Setup interrupt is disabled. |
| 1 | The Transmitted Setup interrupt is enabled. |

**Bit 3 – PERR** Pipe Error Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Pipe Error interrupt.

| Value | Description |
|---|---|
| 0 | The Pipe Error interrupt is disabled. |
| 1 | The Pipe Error interrupt is enabled. |

**Bit 2 – TRFAIL** Transfer Fail Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Fail interrupt.

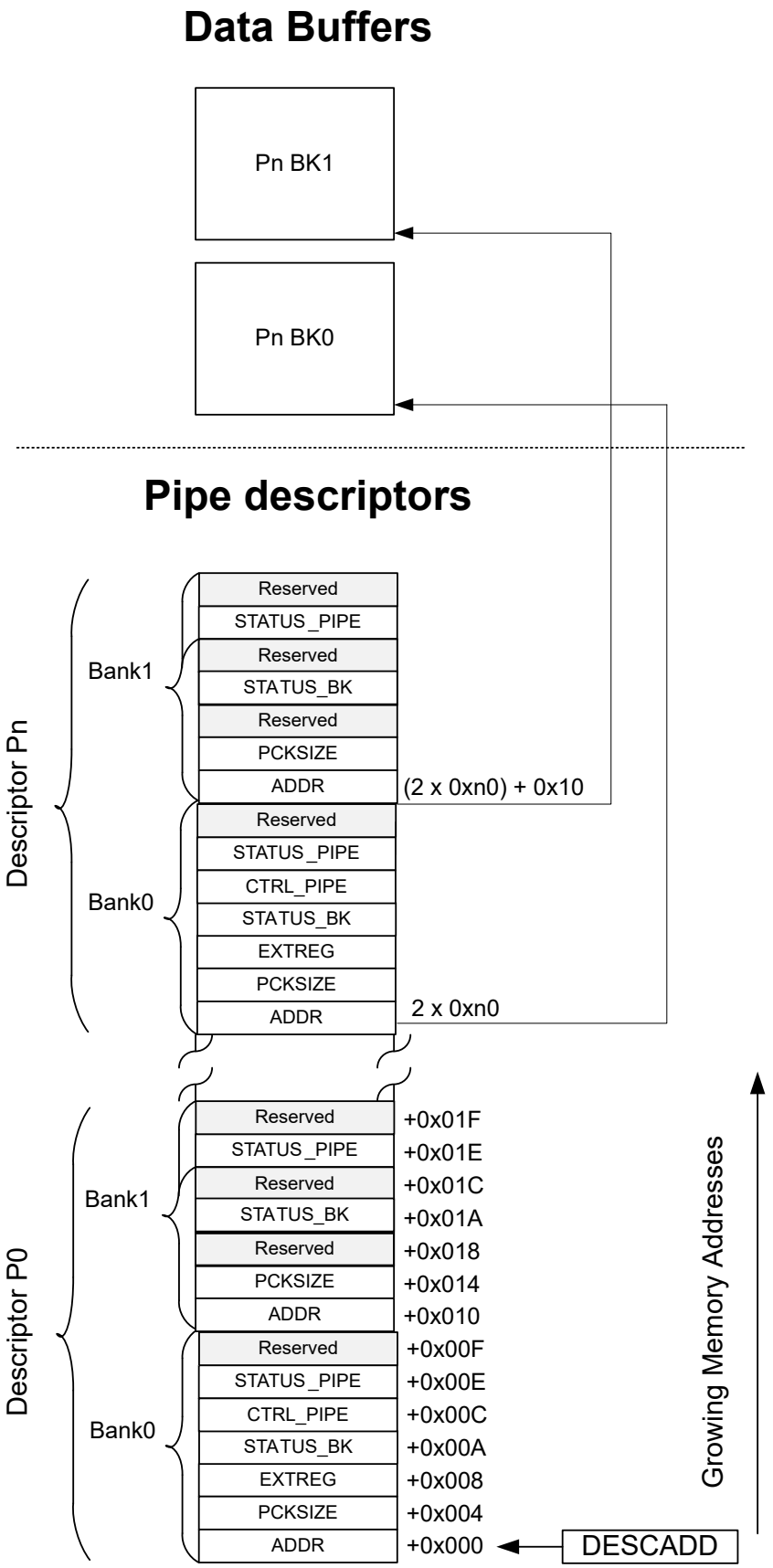
| Value | Description |
|---|---|
| 0 | The Transfer Fail interrupt is disabled. |
| 1 | The Transfer Fail interrupt is enabled. |

**Bits 0, 1 – TRCPT** Transfer Complete x interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will enable the Transfer Complete interrupt Enable bit x.
0.2.7 Host Registers - Pipe RAM

| Value | Description |
|---|---|
| 0 | The Transfer Complete x interrupt is disabled. |
| 1 | The Transfer Complete x interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.14. Pipe Descriptors Structure





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.15. Host Pipe n Descriptor Bank 0 / 1

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x04 | PCKSIZE | 7:0 | BYTE_COUNT[7:0] | | | | | | | |
| | | 15:8 | MULTI_PACKET_SIZE[1:0] | | BYTE_COUNT[13:8] | | | | | |
| | | 23:16 | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| | | 31:24 | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| 0x08 | EXTREG | 7:0 | VARIABLE[3:0] | | | | SUBPID[3:0] | | | |
| | | 15:8 | | VARIABLE[10:4] | | | | | | |
| 0x0A | STATUS_BK | 7:0 | | | | | | | ERRORFLOW | CRCERR |
| 0x0B | Reserved | | | | | | | | | |
| 0x0C | CTRL_PIPE | 7:0 | | PDADDR[6:0] | | | | | | |
| | | 15:8 | PERMAX[3:0] | | | | PEPNUM[3:0] | | | |
| 0x0E | STATUS_PIPE | 7:0 | ERCNT[2:0] | | | CRC16ER | TOUTER | PIDER | DAPIDER | DTGLER |
| | | 15:8 | | | | | | | | |
| 0x10 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x14 | PCKSIZE | 7:0 | BYTE_COUNT[7:0] | | | | | | | |
| | | 15:8 | MULTI_PACKET_SIZE[1:0] | | BYTE_COUNT[13:8] | | | | | |
| | | 23:16 | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| | | 31:24 | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| 0x18 ... 0x19 | Reserved | | | | | | | | | |
| 0x1A | STATUS_BK | 7:0 | | | | | | | ERRORFLOW | CRCERR |
| 0x1B ... 0x1D | Reserved | | | | | | | | | |
| 0x1E | STATUS_PIPE | 7:0 | ERCNT[2:0] | | | CRC16ER | TOUTER | PIDER | DAPIDER | DTGLER |
| | | 15:8 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.15.1. Address of the Data Buffer

**Name:** ADDR
**Offset:** 0x00 + n*0x04 [n=0..1]
**Reset:** 0xxxxxxxx
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADDR[31:0]** Data Pointer Address Value

These bits define the data pointer address as an absolute double word address in RAM. The two least significant bits must be zero to ensure the descriptor is 32-bit aligned.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.15.2. Packet Size

**Name:** PCKSIZE
**Offset:** 0x04 + n*0x04 [n=0..1]
**Reset:** 0xXXXXXXX
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MULTI_PACKET_SIZE[1:0] | | BYTE_COUNT[13:8] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | x | x | x | x | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BYTE_COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – AUTO_ZLP** Automatic Zero Length Packet
This bit defines the automatic Zero Length Packet mode of the pipe.
When enabled, the USB module will manage the ZLP handshake by hardware. This bit is for OUT pipes only. When disabled the handshake must be managed by firmware.

| Value | Description |
|---|---|
| 0 | Automatic Zero Length Packet is disabled. |
| 1 | Automatic Zero Length Packet is enabled. |

**Bits 30:28 – SIZE[2:0]** Pipe size
These bits contains the size of the pipe.
These bits are cleared upon sending a USB reset.

| SIZE[2:0] | Description |
|---|---|
| 0x0 | 8 Byte |
| 0x1 | 16 Byte |
| 0x2 | 32 Byte |
| 0x3 | 64 Byte |
| 0x4 | 128 Byte(1) |
| 0x5 | 256 Byte(1) |
| 0x6 | 512 Byte(1) |
| 0x7 | 1023 Byte(1) |

**Note:**
1. For Isochronous pipe only.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 27:14 – MULTI_PACKET_SIZE[13:0]** Multi Packet IN or OUT size

These bits define the 14-bit value that is used for multi-packet transfers.
For IN pipes, MULTI_PACKET_SIZE holds the total number of bytes sent. MULTI_PACKET_SIZE must be written to zero when setting up a new transfer.
For OUT pipes, MULTI_PACKET_SIZE holds the total data size for the complete transfer. This value must be a multiple of the maximum packet size.

**Bits 13:0 – BYTE_COUNT[13:0]** Byte Count

These bits define the 14-bit value that contains number of bytes sent in the last OUT or SETUP transaction for an OUT pipe, or of the number of bytes to be received in the next IN transaction for an input pipe.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.15.3. Extended Register

**Name:** EXTREG
**Offset:** 0x08
**Reset:** 0xXXXXXXX
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | VARIABLE[10:4] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | x | x | x | x | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VARIABLE[3:0] | | | | SUBPID[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | x | x | x | x |

**Bits 14:4 – VARIABLE[10:0]** Variable field send with extended token
These bits define the VARIABLE field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum.”
To support the USB2.0 Link Power Management addition the VARIABLE field should be set as described below.

| VARIABLE | Description |
|---|---|
| VARIABLE[3:0] | bLinkState(1) |
| VARIABLE[7:4] | BESL (See LPM ECN)(2) |
| VARIABLE[8] | bRemoteWake(1) |
| VARIABLE[10:9] | Reserved |

**Notes:**

1. For a definition of LPM Token bRemoteWake and bLinkState fields, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum".
2. For a definition of LPM Token BESL field, refer to "Table 2-3 in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum" and "Table X-X1 in Errata for ECN USB 2.0 Link Power Management.

**Bits 3:0 – SUBPID[3:0]** SUBPID field send with extended token
These bits define the SUBPID field sent with extended token. See “Section 2.1.1 Protocol Extension Token in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.
To support the USB2.0 Link Power Management addition the SUBPID field should be set as described in “Table 2.2 SubPID Types in the reference document ENGINEERING CHANGE NOTICE, USB 2.0 Link Power Management Addendum”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.15.4. Host Status Bank

**Name:** STATUS_BK
**Offset:** 0x0A + n*0x10 [n=0..1]
**Reset:** 0xXXXXXXX
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ERRORFLOW | CRCERR |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | x | x |

**Bit 1 – ERRORFLOW** Error Flow Status
This bit defines the Error Flow Status.
This bit is set when a Error Flow has been detected during transfer from/towards this bank.
For IN transfer, a NAK handshake has been received. For OUT transfer, a NAK handshake has been received. For Isochronous IN transfer, an overrun condition has occurred. For Isochronous OUT transfer, an underflow condition has occurred.

| Value | Description |
|---|---|
| 0 | No Error Flow detected. |
| 1 | A Error Flow has been detected. |

**Bit 0 – CRCERR** CRC Error
This bit defines the CRC Error Status.
This bit is set when a CRC error has been detected in an isochronous IN endpoint bank.

| Value | Description |
|---|---|
| 0 | No CRC Error. |
| 1 | CRC Error detected. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.15.5. Host Control Pipe

**Name:** CTRL_PIPE
**Offset:** 0x0C
**Reset:** 0xXXXX
**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PERMAX[3:0] | | | | PEPNUM[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | PDADDR[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | x | x | x | x | 0 | 0 | 0 |

**Bits 15:12 – PERMAX[3:0]** Pipe Error Max Number
These bits define the maximum number of error for this Pipe before freezing the pipe automatically.

**Bits 11:8 – PEPNUM[3:0]** Pipe EndPoint Number
These bits define the number of endpoint for this Pipe.

**Bits 6:0 – PDADDR[6:0]** Pipe Device Address
These bits define the Device Address for this pipe.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.15.6. Host Status Pipe

**Name:** STATUS_PIPE
**Offset:** 0x0E + n*0x02 [n=0..1]
**Reset:** 0xXXXXXXX
**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERCNT[2:0] | | | CRC16ER | TOUTER | PIDER | DAPIDER | DTGLER |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | x | x | x | x | x | x | x | x |

**Bits 7:5 – ERCNT[2:0]** Pipe Error Counter
These bits define the number of errors detected on the pipe.

**Bit 4 – CRC16ER** CRC16 ERROR
This bit defines the CRC16 Error Status.
This bit is set when a CRC 16 error has been detected during a IN transactions.

| Value | Description |
|---|---|
| 0 | No CRC 16 Error detected. |
| 1 | A CRC 16 error has been detected. |

**Bit 3 – TOUTER** TIME OUT ERROR
This bit defines the Time Out Error Status.
This bit is set when a Time Out error has been detected during a USB transaction.

| Value | Description |
|---|---|
| 0 | No Time Out Error detected. |
| 1 | A Time Out error has been detected. |

**Bit 2 – PIDER** PID ERROR
This bit defines the PID Error Status.
This bit is set when a PID error has been detected during a USB transaction.

| Value | Description |
|---|---|
| 0 | No PID Error detected. |
| 1 | A PID error has been detected. |

**Bit 1 – DAPIDER** Data PID ERROR
This bit defines the PID Error Status.
This bit is set when a Data PID error has been detected during a USB transaction.

| Value | Description |
|---|---|
| 0 | No Data PID Error detected. |
| 1 | A Data PID error has been detected. |

**Bit 0 – DTGLER** Data Toggle Error
This bit defines the Data Toggle Error Status.
This bit is set when a Data Toggle Error has been detected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | No Data Toggle Error. |
| 1 | Data Toggle Error detected. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 39. CAN - Control Area Network

## 39.1. Overview

The Control Area Network (CAN) performs communication according to ISO 11898-1:2015 (Bosch CAN specification 2.0 part A, B, ISO CAN FD). Message storage is intended to be a single-ported or dual-ported Message RAM outside of the module.

## 39.2. Features

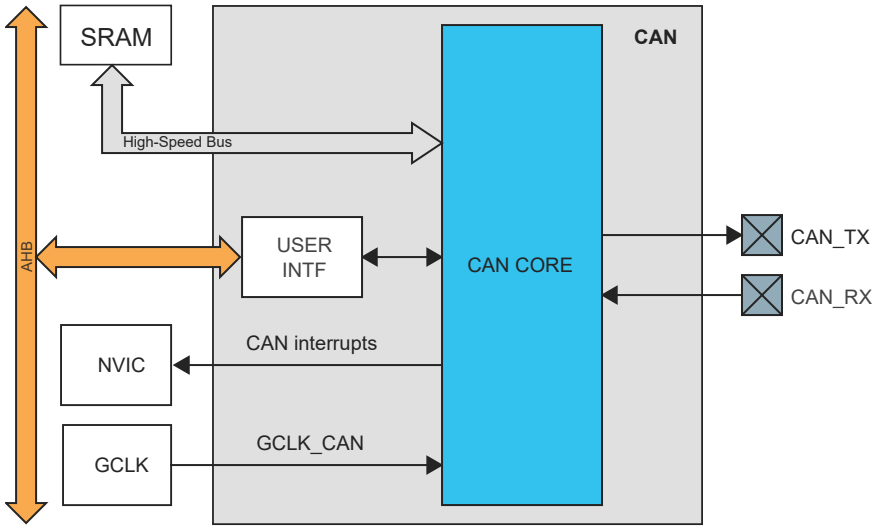
The following are key features of the CAN module:

- Conform with CAN protocol version 2.0 part A and part B, and ISO 11898-1:2015
- Up to two Controller Area Network (CAN) interfaces
  - Supporting CAN 2.0 part A and part B, and CAN-FD (ISO 11898-1:2015)
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- AUTOSAR optimized
- SAE J1939 optimized
- Two configurable Receive FIFOs
- Separate signaling on reception of high-priority messages
- Up to 64 dedicated Receive Buffers and up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO, Transmit Queue, Transmit Event FIFO
- Direct Message RAM access for CPU
- Programmable Loop-Back Test mode
- Maskable module interrupts
- Power-down support; debug on CAN support
- Transfer rates:
  - 1 Mb/s for CAN 2.0 mode
  - 10 Mb/s for CAN-FD mode


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 39.3. Block Diagram

**Figure 39-1.** CAN Block Diagram




## 39.4. Signal Description

**Table 39-1.** Signal Description

| Signal | Description | Type |
|---|---|---|
| CAN_TX | CAN transmit | Digital output |
| CAN_RX | CAN receive | Digital input |

One signal can be mapped to one of several pins. Refer to the Pinout and Packaging chapter for more details.

## 39.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 39.5.1. I/O Lines

Using the CAN's I/O lines requires the I/O pins to be configured.

**Related Links**

PORT - I/O Pin Controller

### 39.5.2. Power Management

The CAN will continue to operate in any Idle Sleep mode where the selected source clock is running. The CAN interrupts can be used to wake up the device from sleep modes. Refer to the Power Manager chapter for details on the different sleep modes.

The CAN module has its own Low-Power mode. The clock sources cannot be halted while the CAN is enabled unless this mode is used. Refer to the section "Sleep Mode Operation" for additional information.

**Related Links**

Sleep Mode Operation


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.5.3. Clocks

An AHB clock (CLK_CAN_AHB) is required to clock the CAN. This clock can be configured in the Main Clock peripheral (MCLK) before using the CAN, and the default state of CLK_CAN_AHB can be found in the MCLK.AHBMASK register.

A generic clock (GCLK_CAN) is required to clock the CAN. This clock must be configured and enabled in the generic clock controller before using the CAN.

This generic clock is asynchronous to the bus clock (CLK_CAN_AHB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

**Related Links**
Peripheral Clock Masking
GCLK - Generic Clock Controller

### 39.5.4. DMA

The CAN has a built-in Direct Memory Access (DMA) and will read/write data to/from the system RAM when a CAN transaction takes place. No CPU or DMA Controller (DMAC) resources are required.

The DMAC can be used for debug messages functionality.

**Related Links**
DMAC – Direct Memory Access Controller

### 39.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the CAN interrupts requires the interrupt controller to be configured first.

### 39.5.6. Events

Not applicable.

### 39.5.7. Debug Operation

Not applicable.

### 39.5.8. Register Access Protection

Not applicable.

### 39.5.9. Analog Connections

No analog connections.

## 39.6. Functional Description

### 39.6.1. Principle of Operation

The CAN performs communication according to ISO 11898-1:2015 (identical to Bosch CAN protocol specification 2.0 part A,B, ISO CAN FD).

The message storage is intended to be a single- or dual-ported Message RAM outside the module. It is connected to the CAN via AHB.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to the Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Acceptance filtering is implemented by a combination of up to 128 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

## 39.6.2. Operating Modes

### 39.6.2.1. Software Initialization

Software initialization is started by setting bit CCCR.INIT, either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off. While CCCR.INIT is set, message transfer from and to the CAN bus is stopped, the status of the CAN bus output CAN_TX is ”recessive” (HIGH). The counters of the Error Management Logic EML are unchanged. Setting CCCR.INIT does not change any configuration register. Resetting CCCR.INIT finishes the software initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive ”recessive” bits (= Bus_Idle) before it can take part in bus activities and start the message transfer.

Access to the CAN configuration registers is only enabled when both bits CCCR.INIT and CCCR.CCE are set (protected write).

CCCR.CCE can only be set/reset while CCCR.INIT = ‘1’. CCCR.CCE is automatically reset when CCCR.INIT is reset.

The following registers are reset when CCCR.CCE is set

- HPMS - High Priority Message Status
- RXF0S - Rx FIFO 0 Status
- RXF1S - Rx FIFO 1 Status
- TXFQS - Tx FIFO/Queue Status
- TXBRP - Tx Buffer Request Pending
- TXBTO - Tx Buffer Transmission Occurred
- TXBCF - Tx Buffer Cancellation Finished
- TXEFS - Tx Event FIFO Status

The Timeout Counter value TOCV.TOC is preset to the value configured by TOCC.TOP when CCCR.CCE is set.

In addition the state machines of the Tx Handler and Rx Handler are held in idle state while CCCR.CCE = ‘1’.

The following registers are only writable while CCCR.CCE = ‘0’

- TXBAR - Tx Buffer Add Request
- TXBCR - Tx Buffer Cancellation Request

CCCR.TEST and CCCR.MON can only be set by the CPU while CCCR.INIT = ‘1’ and CCR.CCE = ‘1’. Both bits may be reset at any time. CCCR.DAR can only be set/reset while CCCR.INIT = ‘1’ and CCCR.CCE = ‘1’.

### 39.6.2.2. Normal Operation

Once the CAN is initialized and CCCR.INIT is reset to ‘0’, the CAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO0 or Rx FIFO1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.6.2.3. CAN FD Operation

There are two variants in the CAN FD frame format, first the CAN FD frame without bit rate switching where the data field of a CAN frame may be longer than 8 bytes. The second variant is the CAN FD frame where control field, data field, and CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the CAN receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit PSR.PXE. When Protocol Exception Handling is enabled (CCCR.PXHD = '0'), this causes the operation state to change from Receiver (PSR.ACT = "10") to synchronizing on CAN communication (PSR.ACT = "00") at the next sample point. In case Protocol Exception Handling is disabled (CCCR.PXHD = '1'), the CAN will treat a recessive res bit as a form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR.FDOE. In case CCCR.FDOE = '1', transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element. With CCCR.FDOE = '0', received frames are interpreted as Classic CAN frames, witch leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set. CCCR.FDOE and CCCR.BRSE can only be changed while CCCR.INIT and CCCR.CCE are both set.

With CCCR.FDOE = '0', the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. With CCCR.FDOE = '1' and CCCR.BRSE = '0', only bit FDF of a Tx Buffer element is evaluated. With CCCR.FDOE = '1' and CCCR.BRSE = '1', transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significantly higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to the table below.

**Table 39-2.** Coding of DLC in CAN FD

| DLC | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|
| Number of Data Bytes | 12 | 16 | 20 | 24 | 32 | 48 | 64 |

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the Nominal Bit Timing & Prescaler Register NBTP. In the following CAN FD data phase, the fast CAN bit timing is used as defined by the Data Bit Timing & Prescaler Register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

DBTP. The bit timing is switched back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency (GCLK_CAN). Example: with a CAN clock frequency of 20MHz and the shortest configurable bit time of 4 $t_q$, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD long and CAN FD fast, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

#### 39.6.2.4. Transceiver Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin CAN_TX the CAN receives the transmitted data from its local CAN transceiver via pin CAN_RX. The received data is delayed by the CAN transceiver's loop delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transceiver loop delay, the delay compensation is introduced. Without transceiver delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transceivers loop delay.

Description

The CAN's protocol unit has implemented a delay compensation mechanism to compensate the transmitter delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point SSP. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting bit DBTP.TDC.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the CAN's transmit output CAN_TX through the transceiver to the receive input CAN_RX plus the transmitter delay compensation offset as configured by TDCR.TDCO. The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of mtq.

PSR.TDCV shows the actual transmitter delay compensation value. PSR.TDCV is cleared when CCCR.INIT is set and is updated at each transmission of an FD frame while DBTP.TDC is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN:

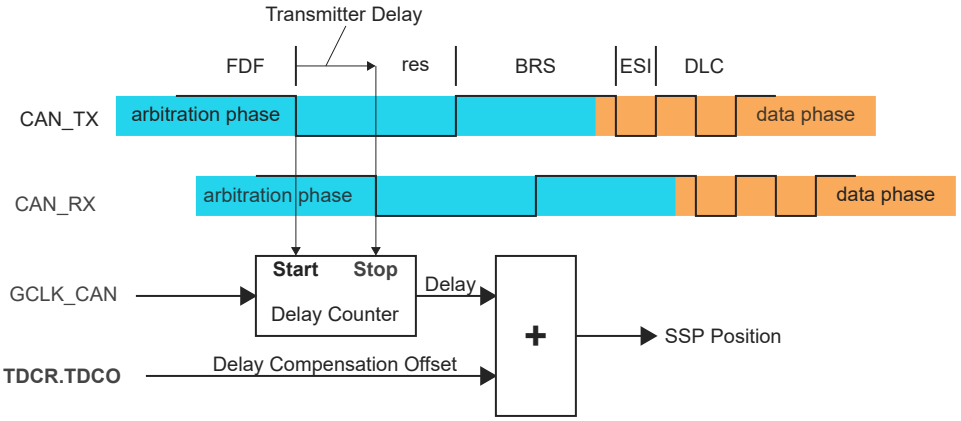
- The sum of the measured delay from CAN_TX to CAN_RX and the configured transceiver delay compensation offset FBTP.TDCO has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CAN_TX to CAN_RX and the configured transceiver delay compensation offset FBTP.TDCO has to be less or equal to 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transceiver delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.
  Transmitter Delay Compensation Measurement

If transmitter delay compensation is enabled by programming DBTP.TDC = '1', the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

measurement is stopped when this edge is seen at the receive input CAN_TX of the transmitter. The resolution of this measurement is one mtq.

**Figure 39-2.** Transceiver delay measurement




To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a too early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming TDCR.TDCF. This defines a minimum value for the SSP position. Dominant edges of CAN_RX, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCR.TDCF AND CAN _RX is low.

### 39.6.2.5. Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (ECR.REC, ECR.TEC) are frozen while Error Logging (ECR.CEL) is still incremented. The CPU can set the CAN into Restricted Operation mode by setting bit CCCR.ASM. The bit can only be set by the CPU when both CCCR.CCE and CCCR.INIT are set to '1'. The bit can be reset by the CPU at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the CPU has to reset CCCR.ASM.

The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

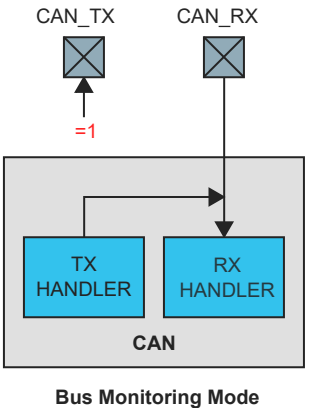
### 39.6.2.6. Bus Monitoring Mode

The CAN is set in Bus Monitoring Mode by programming CCCR.MON to '1'. In Bus Monitoring Mode (see ISO 11898-1, 10.12 Bus monitoring), the CAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the CAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the CAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode register TXBRP is held in reset state.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. The figure below shows the connection of signals CAN_TX and CAN_RX to the CAN in Bus Monitoring Mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 39-3.** Pin Control in Bus Monitoring Mode




### 39.6.2.7. Disabled Automatic Retransmission

According to the CAN Specification (see ISO 11898-1, 6.3.3 Recovery Management), the CAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via CCCR.DAR.

Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending bit TXBRP.TRPx is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx not set
- Successful transmission in spite of cancellation:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set
- Arbitration lost or frame transmission disturbed:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO.TOx not set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF.CFx set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = "10" (transmission in spite of cancellation).

### 39.6.2.8. Test Modes

To enable write access to the TEST register, the CCCR bit has to be set to '1'. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin (CANx_TX) by programming TEST.TX. Additionally to its default function, the serial data output, it can drive the CAN Sample Point signal to monitor the CAN's bit timing and it can drive constant dominant or recessive values. The actual value at the CANx_RX pin can be read from TEST.RX. Both functions can be used to check the CAN bus physical layer.

Due to the synchronization mechanism between GCLK_CANx and GCLK_CANx_AHB domains, there may be a delay of several GCLK_CANx_AHB periods between writing to TEST.TX until the new


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

configuration is visible at output pin CANx_TX. This also applies when reading the CANx_RX input pin through TEST.RX.

**Note:** Test modes must be used for production tests or self test only. The software control for the CANx_TX pin interferes with all CAN protocol functions. It is not recommended to use test modes for application.

### External Loop-Back Mode

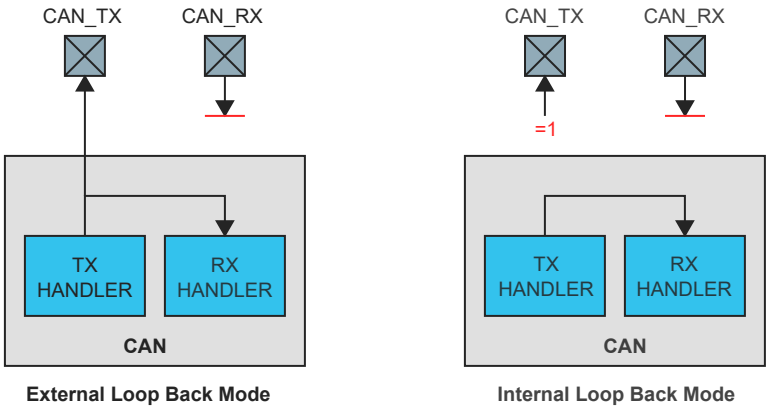
The CAN can be set in External loop-back mode by programming TEST.LBCK to '1'. In loop-back mode, the CAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The figure below shows the connection of CANx_TX and CANx_RX signals to the CAN in external loop-back mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the CAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in loop-back mode. In this mode the CAN performs an internal feedback from its Tx output to its Rx input. The actual value of the CANx_RX input pin is disregarded by the CAN. The transmitted messages can be monitored at the CANx_TX pin.

### Internal Loop-Back Mode

Internal loop-back mode is entered by programming the TEST.LBCK and CCCR.MON bits to '1'. This mode can be used for a 'Hot Selftest', meaning the CAN can be tested without affecting a running CAN system connected to the CANx_TX and CANx_RX pins. In this mode, the CANx_RX pin is disconnected from the CAN and the CANx_TX pin is held recessive. The figure below shows the connection of the CANx_TX and CANx_RX pins to the CAN in case of internal loop-back mode.

**Figure 39-4.** Pin Control in Loop-Back Modes




## 39.6.3. Timestamp Generation

For timestamp generation, the CAN supplies a 16-bit wrap-around counter. A prescaler TSCC.TCP can be configured to clock the counter in multiples of CAN bit times (1...16). The counter is readable through TSCV.TSC. A write access to TSCV register resets the counter to '0'. When the timestamp counter wraps around interrupt flag IR.TSW is set.

On start of frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer/Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

**Note:** In CANFD mode, if the bit rate switching is turned on (that is, CCCR.BRSE = 1), this counter can not be used.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.6.4. Timeout Counter

To indicate timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO the CAN supplies a 16-bit timeout counter. It operates as down-counter and uses the same prescaler controlled by TSCC.TCP as the timestamp counter. The timeout counter is configured through the TOCC register. The actual counter value can be read from TOCV.TOC. The timeout counter can only be started while CCCR.INIT = '0'. It is stopped when CCCR.INIT = '1', for example, when the CAN enters Bus_Off state.

The operation mode is selected by TOCC.TOS. When operating in continuous mode, the counter starts when CCCR.INIT is reset. A write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting.

When the timeout counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches zero, interrupt flag IR.TOO is set. In continuous mode, the counter is immediately restarted at TOCC.TOP.

**Note:** The clock signal for the timeout counter is derived from the CAN Core's sample point signal. Therefore the point in time where the timeout counter is decremented may vary due to the synchronization/resynchronization mechanism of the CAN core. If the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

### 39.6.5. Rx Handling

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

#### 39.6.5.1. Acceptance Filtering

The CAN offers the possibility to configure two sets of acceptance filters: one for standard identifiers and one for extended identifiers. These filters can be assigned to an Rx Buffer or to Rx FIFO 0, 1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are as follows:

- Each filter element can be configured as
  - Range filter (from - to)
  - Filter for one or two dedicated IDs
  - Classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled/disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

The related configuration registers are listed below:

- Global Filter Configuration (GFC)
- Standard ID Filter Configuration (SIDFC)
- Extended ID Filter Configuration (XIDFC)
- Extended ID AND Mask (XIDAM)

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx buffer


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Store received frame in Rx buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM
- Set High Priority Message interrupt flag IR.HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx buffer or Rx FIFO. If the CAN protocol controller has detected an error condition, for example CRC error, this message is discarded with the following impact on the affected Rx buffer or Rx FIFO:

Rx Buffer
– New Data flag of matching Rx buffer is not set, but Rx buffer (partly) overwritten with received data. For error type, see PSR.LEC respectively PSR.FLEC.

Rx FIFO
– Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type, see PSR.LEC respectively PSR.FLEC. In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in Rx FIFO Overwrite Mode must be considered.

**Note:** When an accepted message is written to one of the two Rx FIFOs or into an Rx buffer, the unmodified received identifier is stored independent of the filters used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

**Range Filter**

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID for standard frames or EF1ID/EF2ID for extended frames.

There are two possibilities when range filtering is used together with extended frames:

| | |
|---|---|
| **EFT = "00"** | The Message ID of received frames is AND'ed with the Extended ID AND Mask (XIDAM) before the range filter is applied |
| **EFT = "11"** | The Extended ID AND Mask (XIDAM) is not used for range filtering |

**Filter for specific IDs**

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

**Classic Bit Mask Filter**

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, for example, the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

If all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

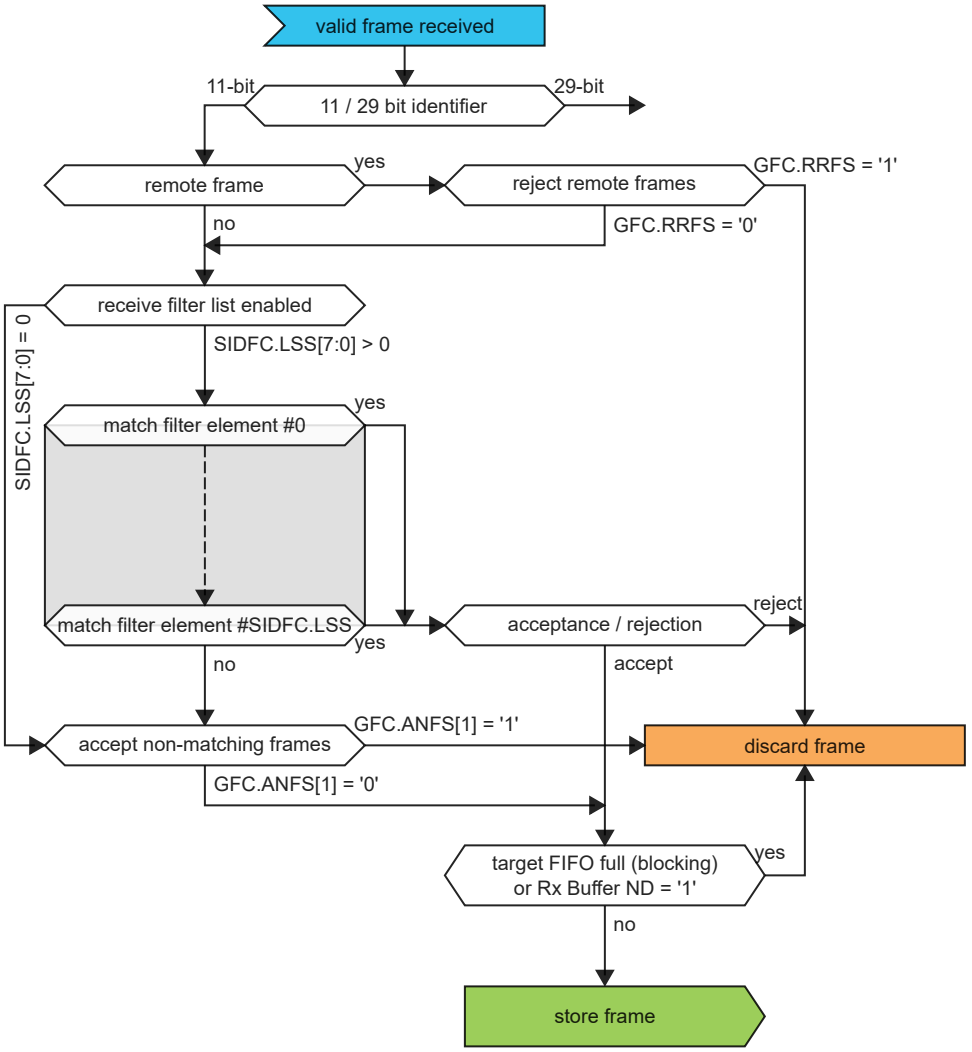
**Standard Message ID Filtering**

The figure below shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in Standard Message ID Filter Element.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Controlled by the Global Filter Configuration (GFC) and the Standard ID Filter Configuration (SIDFC) Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

**Figure 39-5.** Standard Message ID Filtering




### Extended Message ID Filtering

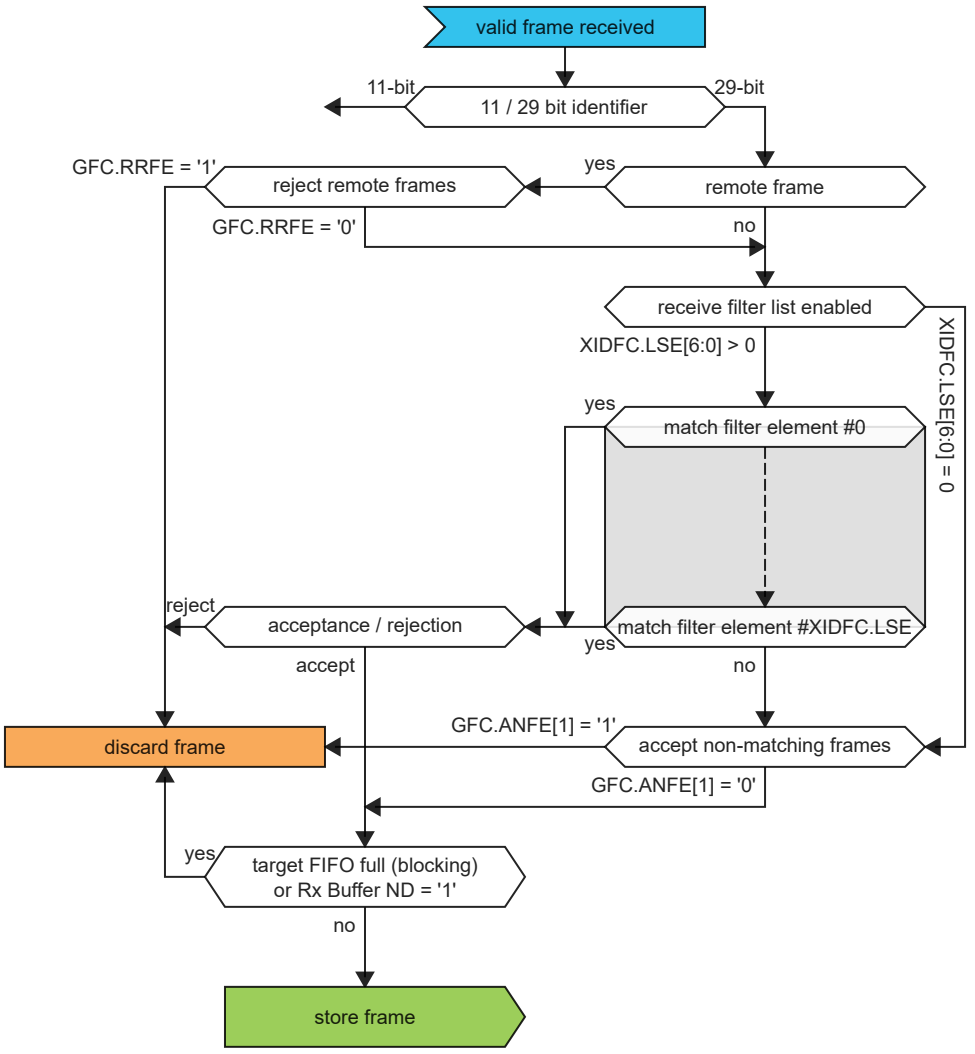
The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in Extended Message ID Filter Element.

Controlled by the Global Filter Configuration (GFC) and the Extended ID Filter Configuration (XIDFC) Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask (XIDAM) is AND'ed with the received identifier before the filter list is executed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 39-6.** Extended Message ID Filtering




### 39.6.5.2. Rx FIFOs

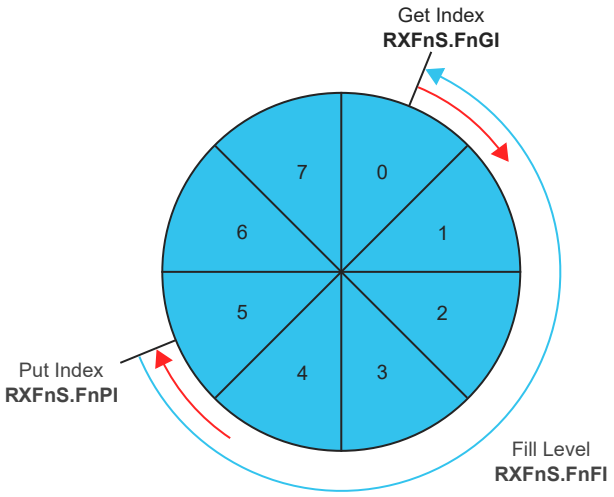
Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done through the RXF0C and RXF1C registers.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1, refer to the Acceptance Filtering. The Rx FIFO element is described in Rx Buffer and FIFO Element.

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by RXFnC.FnWM, interrupt flag IR.RFnW is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signalled by RXFnS.FnF. In addition interrupt flag IR.RFnF is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 39-7.** Rx FIFO Status




When reading from an Rx FIFO, Rx FIFO Get Index RXFnS.FnGI • FIFO Element Size has to be added to the corresponding Rx FIFO start address RXFnC.FnSA.

**Table 39-3.** Rx Buffer/FIFO Element Size

| RXESC.RBDS[2:0]<br>RXESC.FnDS[2:0] | Data Field<br>[bytes] | FIFO Element Size<br>[RAM words] |
|---|---|---|
| 000 | 8 | 4 |
| 001 | 12 | 5 |
| 010 | 16 | 6 |
| 011 | 20 | 7 |
| 100 | 24 | 8 |
| 101 | 32 | 10 |
| 110 | 48 | 14 |
| 111 | 64 | 18 |

### Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by RXFnC.FnOM = '0'. This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached (RXFnS.FnPI = RXFnS.FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signaled by RXFnS.FnF = '1'. In addition interrupt flag IR.RFnF is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by RXFnS.RFnL = '1'. In addition interrupt flag IR.RFnL is set.

### Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by RXFnC.FnOM = '1'.

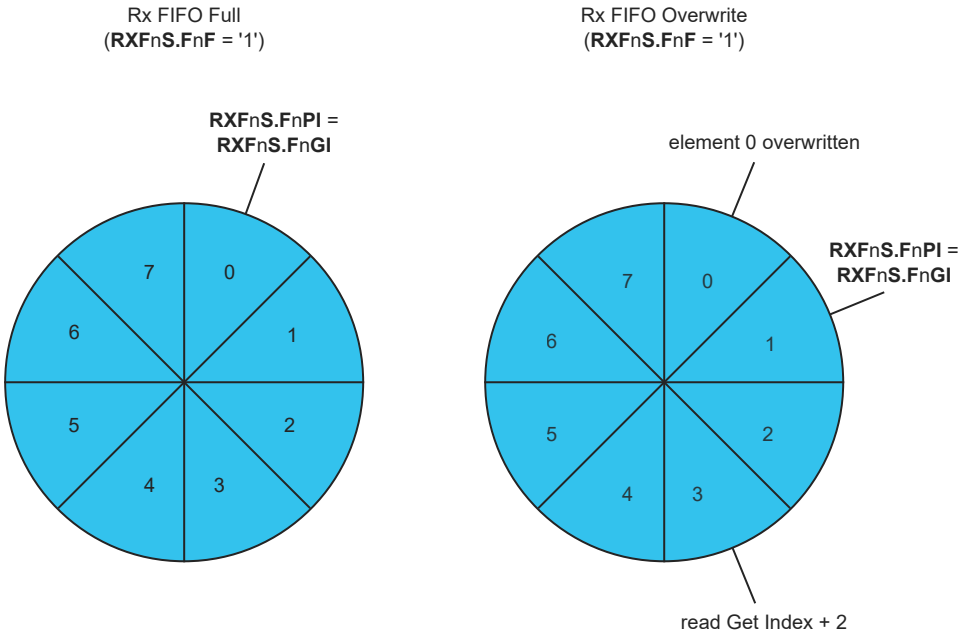
When an Rx FIFO full condition (RXFnS.FnPI = RXFnS.FnGI) is signaled by RXFnS.FnF = '1', the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signaled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. The figure below shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

**Figure 39-8.** Rx FIFO Overflow Handling




After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index RXFnA.FnA. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (RXFnS.FnF = '0').

### 39.6.5.3. Dedicated Rx Buffers

The CAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via RXBC.RBSA.

For each Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = "111" and SFID2 / EFID2[10:9] = "00" has to be configured (see Standard Message ID Filter Element and Extended Message ID Filter Element).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag IR.DRX (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

**Table 39-4.** Example Filter Configuration for Rx Buffers

| Filter Element | SFID1[10:0] / EFID1[28:0] | SFID2[10:9] / EFID2[10:9] | SFID2[5:0] / EFID2[5:0] |
|---|---|---|---|
| 0 | ID message 1 | 00 | 00 0000 |
| 1 | ID message 2 | 00 | 00 0001 |
| 2 | ID message 3 | 00 | 00 0010 |

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1, NDAT2 is set. As long as the New Data flag is set, the


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the CPU by writing a '1' to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

**Rx Buffer Handling**

- Reset interrupt flag IR.DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

### 39.6.5.4. Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element (see Rx Buffer and FIFO Element ).

Advantage: Fixed start address for the DMA transfers (relative to RXBC.RBSA), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = "111" have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the CAN while DMA request is activated. The behavior is similar to that of an Rx Buffers with its New Data flag set.

After the DMA has completed the DMA unit sets the DMA acknowledge. This resets DMA request. Now the CAN is prepared to receive the next set of debug messages.

**Filtering for Debug Messages**

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be programmed to "111". In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning (see Standard Message ID Filter Element and Extended Message ID Filter Element). While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor IR.DRX are set. The reception of debug messages can be monitored via RXF1S.DMS.

**Table 39-5.** Example Filter Configuration for Debug Messages

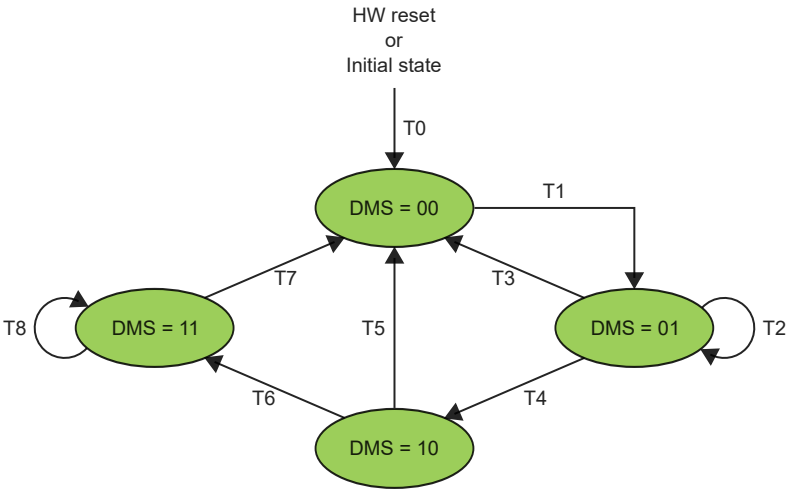
| Filter Element | SFID1[10:0] / EFID1[28:0] | SFID2[10:9] / EFID2[10:9] | SFID2[5:0] / EFID2[5:0] |
|---|---|---|---|
| 0 | ID debug message A | 01 | 11 1101 |
| 1 | ID debug message B | 10 | 11 1110 |
| 2 | ID debug message C | 11 | 11 1111 |

**Debug Message Handling**

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 39-9.** Debug Message Handling State Machine




T0: Reset DMA request output, enable reception of debug message A, B, and C
T1: Reception of debug message A
T2: Reception of debug message A
T3: Reception of debug message C
T4: Reception of debug message B
T5: Reception of debug message A, B
T6: Reception of debug message C
T7: DMA transfer completed
T8: Reception of debug message A, B, C (message rejected)

### 39.6.6. Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in Tx Buffer Element. The table below describes the possible configurations for frame transmission.

**Table 39-6.** Possible Configurations for Frame Transmission

| CCCR | | Tx Buffer Element | | Frame Transmission |
|---|---|---|---|---|
| BRSE | FDOE | FDF | BRS | |
| ignored | 0 | ignored | ignored | Classic CAN |
| 0 | 1 | 0 | ignored | Classic CAN |
| 0 | 1 | 1 | ignored | FD without bit rate switching |
| 1 | 1 | 0 | ignored | Classic CAN |
| 1 | 1 | 1 | 0 | FD without bit rate switching |
| 1 | 1 | 1 | 1 | FD with bit rate switching |

Note: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

#### 39.6.6.1. Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If e.g. CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit CCCR.TXP. If the bit is set, the CAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR.TXP = '0').

This feature looses up burst transmissions coming from a single node and it protects against "babbling idiot" scenarios where the application program erroneously requests too many transmissions.

#### 39.6.6.2. Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR.ARn. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (refer to table below). Therefore the start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0...31) • Element Size to the Tx Buffer Start Address TXBC.TBSA.

**Table 39-7.** Tx Buffer / FIFO / Queue Element Size

| TXESC.TBDS[2:0] | Data Field [bytes] | Element Size [RAM words] |
|---|---|---|
| 000 | 8 | 4 |
| 001 | 12 | 5 |
| 010 | 16 | 6 |
| 011 | 20 | 7 |
| 100 | 24 | 8 |
| 101 | 32 | 10 |
| 110 | 48 | 14 |
| 111 | 64 | 18 |

#### 39.6.6.3. Tx FIFO

Tx FIFO operation is configured by programming TXBC.TFQM to '0'. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index TXFQS.TFGI. After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The CAN calculates the Tx FIFO Free Level TXFQS.TFFL as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index TXFQS.TFQPI. An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS.TFQF = '1') is signaled. In


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a '1' to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is canceled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (refer to Table 39-7). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI (0...31) • Element Size to the Tx Buffer Start Address TXBC.TBSA.

#### 39.6.6.4. Tx Queue

Tx Queue operation is configured by programming TXBC.TFQM to '1'. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TXFQS.TFQPI. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS.TFQF = '1'), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been canceled.

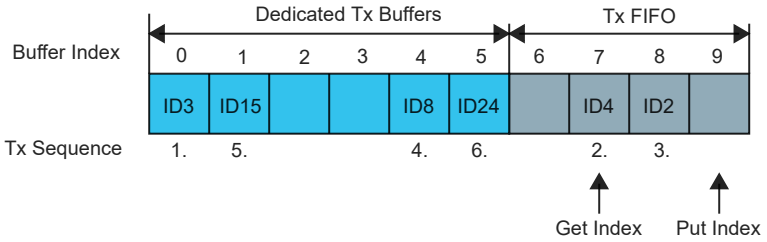
The application may use register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (refer to Table 39-7). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI (0...31) • Element Size to the Tx Buffer Start Address TXBC.TBSA.

#### 39.6.6.5. Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO. The number of Dedicated Tx Buffers is configured by TXBC.NDTB. The number of Tx Buffers assigned to the Tx FIFO is configured by TXBC.TFQS. In case TXBC.TFQS is programmed to zero, only Dedicated Tx Buffers are used.

**Figure 39-10.** Example of mixed Configuration Dedicated Tx Buffers / Tx FIFO




Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TXFS.TFGI)

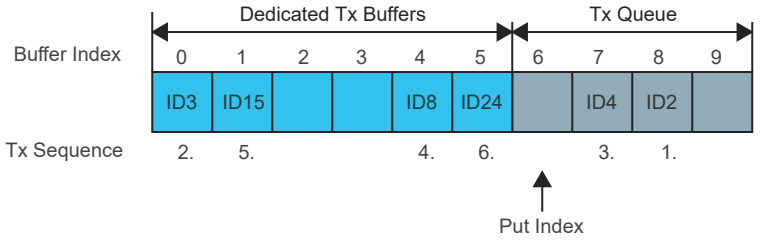

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Buffer with lowest Message ID gets highest priority and is transmitted next

### 39.6.6.6. Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue. The number of Dedicated Tx Buffers is configured by TXBC.NDTB. The number of Tx Queue Buffers is configured by TXBC.TFQS. In case TXBC.TFQS is programmed to zero, only Dedicated Tx Buffers are used.

**Figure 39-11.** Example of mixed Configuration Dedicated Tx Buffers / Tx Queue




Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

### 39.6.6.7. Transmit Cancellation

The CAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the CPU has to write a '1' to the corresponding bit position (=number of Tx Buffer) of register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of register TXBCF to '1'.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding TXBTO and TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding TXBCF bit is set.

**Note:** In case a pending transmission is canceled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

### 39.6.6.8. Tx Event Handling

To support Tx event handling the CAN has implemented a Tx Event FIFO. After the CAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in Tx Event FIFO Element.

When a Tx Event FIFO full condition is signaled by IR.TEFF, no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag IR.TEFL is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by TXEFC.EFWM, interrupt flag IR.TEFW is set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index TXEFS.EFGI has to be added to the Tx Event FIFO start address TXEFC.EFSA.

### 39.6.7. FIFO Acknowledge Handling

The Get Indexes of Rx FIFO 0, Rx FIFO 1 and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (refer to RXF0A, RXF1A and TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.

When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the CAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

**Note:** The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The CAN does not check for erroneous values.

### 39.6.8. Interrupts

The CAN has the following interrupt sources:

- Access to Reserved Address
- Protocol Errors (Data Phase/Arbitration Phase)
- Watchdog Interrupt
- Bus_Off Status
- Error Warning and Passive
- Error Logging Overflow
- Message RAM Bit Errors (Uncorrected/Corrected)
- Message stored to Dedicated Rx Buffer
- Timeout Occurred
- Message RAM Access Failure
- Timestamp Wraparound
- Tx Event FIFO statuses (Element Lost/Full/Watermark Reached/New Entry)
- Tx FIFO Empty
- Transmission Cancellation Finished
- Timestamp Completed
- High Priority Message
- Rx FIFO 1 Statuses (Message Lost/Full/Watermark Reached/New Message)
- Rx FIFO 0 Statuses (Message Lost/Full/Watermark Reached/New Message)

Each interrupt source has an interrupt flag associated with it. The interrupt flag register (IR) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing '1' or disabled by writing '0' to the corresponding bit in the interrupt enable register (IE). Each interrupt flag can be assigned to one of two interrupt service lines.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

An interrupt request is generated when an interrupt flag is set, the corresponding interrupt enable is set, and the corresponding service line enable assigned to the interrupt is set. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, the service line is disabled, or the CAN is reset. Refer to IR for details on how to clear interrupt flags. All interrupt requests from the peripheral are sent to the NVIC. The user must read the IR register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

### 39.6.9. Sleep Mode Operation

The CAN can be configured to operate in any Idle sleep mode. The CAN cannot operate in Standby sleep mode.

The CAN has its own low-power mode that may be used at any time without disabling the CAN. It is also mandatory to allow the CAN to complete all pending transactions before entering Standby mode by activating this low-power mode. This is performed by writing one to the Clock Stop Request bit in the CC Control register (CCCR.CSR = 1). Once all pending transactions are completed and the idle bus state is detected, the CAN will automatically set the Clock Stop Acknowledge bit (CCCR.CSA = 1). The CAN then reverts back to its initial state (CCCR.INIT = 1), blocking further transfers, and it is now safe for CLK_CANx_AHB and GCLK_CANx to be switched off and the system may go to standby.

To leave low-power mode, CLK_CANx_AHB and GCLK_CANx must be active before writing CCCR.CSR to '0'. The CAN will acknowledge this by resetting CCCR.CSA = 0. Afterward, the application can restart CAN communication by resetting the CCCR.INIT bit.

### 39.6.10. Synchronization

Due to the asynchronicity between the main clock domain (CLK_CANx_AHB) and the peripheral clock domain (GCLK_CAN) some registers are synchronized when written. When a write-synchronized register is written, the read back value will not be updated until the register has completed synchronization.

The following bits and registers are write-synchronized: The INIT Initialization bit in the CC Control register (CCCR.INIT).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 39.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CREL | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | SUBSTEP[3:0] | | | | | | | |
| | | 31:24 | REL[3:0] | | | | STEP[3:0] | | | |
| 0x04 | ENDN | 7:0 | ETV[7:0] | | | | | | | |
| | | 15:8 | ETV[15:8] | | | | | | | |
| | | 23:16 | ETV[23:16] | | | | | | | |
| | | 31:24 | ETV[31:24] | | | | | | | |
| 0x08 | MRCFG | 7:0 | | | | | | | QOS[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | DBTP | 7:0 | DTSEG2[3:0] | | | | DSJW[3:0] | | | |
| | | 15:8 | | | | DTSEG1[4:0] | | | | |
| | | 23:16 | TDC | | | DBRP[4:0] | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | TEST | 7:0 | RX | TX[1:0] | | LBCK | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | RWD | 7:0 | WDC[7:0] | | | | | | | |
| | | 15:8 | WDV[7:0] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | CCCR | 7:0 | TEST | DAR | MON | CSR | CSA | ASM | CCE | INIT |
| | | 15:8 | | TXP | EFBI | PXHD | | | BRSE | FDOE |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | NBTP | 7:0 | | NTSEG2[6:0] | | | | | | |
| | | 15:8 | NTSEG1[7:0] | | | | | | | |
| | | 23:16 | NBRP[7:0] | | | | | | | |
| | | 31:24 | NSJW[6:0] | | | | | | | NBRP[8] |
| 0x20 | TSCC | 7:0 | | | | | | | TSS[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | TCP[3:0] | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | TSCV | 7:0 | TSC[7:0] | | | | | | | |
| | | 15:8 | TSC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | TOCC | 7:0 | | | | | | TOS[1:0] | | ETOC |
| | | 15:8 | | | | | | | | |
| | | 23:16 | TOP[7:0] | | | | | | | |
| | | 31:24 | TOP[15:8] | | | | | | | |
| 0x2C | TOCV | 7:0 | TOC[7:0] | | | | | | | |
| | | 15:8 | TOC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x30 ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | ECR | 7:0 | TEC[7:0] | | | | | | | |
| | | 15:8 | RP | REC[6:0] | | | | | | |
| | | 23:16 | CEL[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x44 | PSR | 7:0 | BO | EW | EP | ACT[1:0] | | LEC[2:0] | | |
| | | 15:8 | | PXE | RFDF | RBRS | RESI | DLEC[2:0] | | |
| | | 23:16 | | TDCV[6:0] | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x48 | TDCR | 7:0 | | | | | TDCF[6:0] | | | |
| | | 15:8 | | | | | TDCO[6:0] | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C ... 0x4F | Reserved | | | | | | | | | |
| 0x50 | IR | 7:0 | RF1L | RF1F | RF1W | RF1N | RF0L | RF0F | RF0W | RF0N |
| | | 15:8 | TEFL | TEFF | TEFW | TEFN | TFE | TCF | TC | HPM |
| | | 23:16 | EP | ELO | BEU | BEC | DRX | TOO | MRAF | TSW |
| | | 31:24 | | | ARA | PED | PEA | WDI | BO | EW |
| 0x54 | IE | 7:0 | RF1LE | RF1FE | RF1WE | RF1NE | RF0LE | RF0FE | RF0WE | RF0NE |
| | | 15:8 | TEFLE | TEFFE | TEFWE | TEFNE | TFEE | TCFE | TCE | HPME |
| | | 23:16 | EPE | ELOE | BEUE | BECE | DRXE | TOOE | MRAFE | TSWE |
| | | 31:24 | | | ARAE | PEDE | PEAE | WDIE | BOE | EWE |
| 0x58 | ILS | 7:0 | RF1LL | RF1FL | RF1WL | RF1NL | RF0LL | RF0FL | RF0WL | RF0NL |
| | | 15:8 | TEFLL | TEFFL | TEFWL | TEFNL | TFEL | TCFL | TCL | HPML |
| | | 23:16 | EPL | ELOL | BEUL | BECL | DRXL | TOOL | MRAFL | TSWL |
| | | 31:24 | | | ARAL | PEDL | PEAL | WDIL | BOL | EWL |
| 0x5C | ILE | 7:0 | | | | | | | EINT1 | EINT0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x60 ... 0x7F | Reserved | | | | | | | | | |
| 0x80 | GFC | 7:0 | | | ANFS[1:0] | | ANFE[1:0] | | RRFS | RRFE |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x84 | SIDFC | 7:0 | FLSSA[7:0] | | | | | | | |
| | | 15:8 | FLSSA[15:8] | | | | | | | |
| | | 23:16 | LSS[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x88 | XIDFC | 7:0 | FLESA[7:0] | | | | | | | |
| | | 15:8 | FLESA[15:8] | | | | | | | |
| | | 23:16 | | LSE[6:0] | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x8C ... 0x8F | Reserved | | | | | | | | | |
| 0x90 | XIDAM | 7:0 | EIDM[7:0] | | | | | | | |
| | | 15:8 | EIDM[15:8] | | | | | | | |
| | | 23:16 | EIDM[23:16] | | | | | | | |
| | | 31:24 | | | | EIDM[28:24] | | | | |
| 0x94 | HPMS | 7:0 | MSI[1:0] | | BIDX[5:0] | | | | | |
| | | 15:8 | FLST | FIDX[6:0] | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x98 | NDAT1 | 7:0 | ND7 | ND6 | ND5 | ND4 | ND3 | ND2 | ND1 | ND0 |
| | | 15:8 | ND15 | ND14 | ND13 | ND12 | ND11 | ND10 | ND9 | ND8 |
| | | 23:16 | ND23 | ND22 | ND21 | ND20 | ND19 | ND18 | ND17 | ND16 |
| | | 31:24 | ND31 | ND30 | ND29 | ND28 | ND27 | ND26 | ND25 | ND24 |
| 0x9C | NDAT2 | 7:0 | ND39 | ND38 | ND37 | ND36 | ND35 | ND34 | ND33 | ND32 |
| | | 15:8 | ND47 | ND46 | ND45 | ND44 | ND43 | ND42 | ND41 | ND40 |
| | | 23:16 | ND55 | ND54 | ND53 | ND52 | ND51 | ND50 | ND49 | ND48 |
| | | 31:24 | ND63 | ND62 | ND61 | ND60 | ND59 | ND58 | ND57 | ND56 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xA0 | RXF0C | 7:0 | F0SA[7:0] | | | | | | | |
| | | 15:8 | F0SA[15:8] | | | | | | | |
| | | 23:16 | | F0S[6:0] | | | | | | |
| | | 31:24 | F0OM | F0WM[6:0] | | | | | | |
| 0xA4 | RXF0S | 7:0 | | F0FL[6:0] | | | | | | |
| | | 15:8 | | | F0GI[5:0] | | | | | |
| | | 23:16 | | | F0PI[5:0] | | | | | |
| | | 31:24 | | | | | | | RF0L | F0F |
| 0xA8 | RXF0A | 7:0 | | | F0AI[5:0] | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xAC | RXBC | 7:0 | RBSA[7:0] | | | | | | | |
| | | 15:8 | RBSA[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xB0 | RXF1C | 7:0 | F1SA[7:0] | | | | | | | |
| | | 15:8 | F1SA[15:8] | | | | | | | |
| | | 23:16 | | F1S[6:0] | | | | | | |
| | | 31:24 | F1OM | F1WM[6:0] | | | | | | |
| 0xB4 | RXF1S | 7:0 | | F1FL[6:0] | | | | | | |
| | | 15:8 | | | F1GI[5:0] | | | | | |
| | | 23:16 | | | F1PI[5:0] | | | | | |
| | | 31:24 | DMS[1:0] | | | | | | RF1L | F1F |
| 0xB8 | RXF1A | 7:0 | | | F1AI[5:0] | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xBC | RXESC | 7:0 | | F1DS[2:0] | | | | F0DS[2:0] | | |
| | | 15:8 | | | | | | RBDS[2:0] | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xC0 | TXBC | 7:0 | TBSA[7:0] | | | | | | | |
| | | 15:8 | TBSA[15:8] | | | | | | | |
| | | 23:16 | | | NDTB[5:0] | | | | | |
| | | 31:24 | | TFQM | TFQS[5:0] | | | | | |
| 0xC4 | TXFQS | 7:0 | | | TFFL[5:0] | | | | | |
| | | 15:8 | | | | TFGI[4:0] | | | | |
| | | 23:16 | | | TFQF | TFQPI[4:0] | | | | |
| | | 31:24 | | | | | | | | |
| 0xC8 | TXESC | 7:0 | | | | | | TBDS[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xCC | TXBRP | 7:0 | TRP7 | TRP6 | TRP5 | TRP4 | TRP3 | TRP2 | TRP1 | TRP0 |
| | | 15:8 | TRP15 | TRP14 | TRP13 | TRP12 | TRP11 | TRP10 | TRP9 | TRP8 |
| | | 23:16 | TRP23 | TRP22 | TRP21 | TRP20 | TRP19 | TRP18 | TRP17 | TRP16 |
| | | 31:24 | TRP31 | TRP30 | TRP29 | TRP28 | TRP27 | TRP26 | TRP25 | TRP24 |
| 0xD0 | TXBAR | 7:0 | AR7 | AR6 | AR5 | AR4 | AR3 | AR2 | AR1 | AR0 |
| | | 15:8 | AR15 | AR14 | AR13 | AR12 | AR11 | AR10 | AR9 | AR8 |
| | | 23:16 | AR23 | AR22 | AR21 | AR20 | AR19 | AR18 | AR17 | AR16 |
| | | 31:24 | AR31 | AR30 | AR29 | AR28 | AR27 | AR26 | AR25 | AR24 |
| 0xD4 | TXBCR | 7:0 | CR7 | CR6 | CR5 | CR4 | CR3 | CR2 | CR1 | CR0 |
| | | 15:8 | CR15 | CR14 | CR13 | CR12 | CR11 | CR10 | CR9 | CR8 |
| | | 23:16 | CR23 | CR22 | CR21 | CR20 | CR19 | CR18 | CR17 | CR16 |
| | | 31:24 | CR31 | CR30 | CR29 | CR28 | CR27 | CR26 | CR25 | CR24 |
| 0xD8 | TXBTO | 7:0 | TO7 | TO6 | TO5 | TO4 | TO3 | TO2 | TO1 | TO0 |
| | | 15:8 | TO15 | TO14 | TO13 | TO12 | TO11 | TO10 | TO9 | TO8 |
| | | 23:16 | TO23 | TO22 | TO21 | TO20 | TO19 | TO18 | TO17 | TO16 |
| | | 31:24 | TO31 | TO30 | TO29 | TO28 | TO27 | TO26 | TO25 | TO24 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xDC | TXBCF | 7:0 | CF7 | CF6 | CF5 | CF4 | CF3 | CF2 | CF1 | CF0 |
| | | 15:8 | CF15 | CF14 | CF13 | CF12 | CF11 | CF10 | CF9 | CF8 |
| | | 23:16 | CF23 | CF22 | CF21 | CF20 | CF19 | CF18 | CF17 | CF16 |
| | | 31:24 | CF31 | CF30 | CF29 | CF28 | CF27 | CF26 | CF25 | CF24 |
| 0xE0 | TXBTIE | 7:0 | TIE7 | TIE6 | TIE5 | TIE4 | TIE3 | TIE2 | TIE1 | TIE0 |
| | | 15:8 | TIE15 | TIE14 | TIE13 | TIE12 | TIE11 | TIE10 | TIE9 | TIE8 |
| | | 23:16 | TIE23 | TIE22 | TIE21 | TIE20 | TIE19 | TIE18 | TIE17 | TIE16 |
| | | 31:24 | TIE31 | TIE30 | TIE29 | TIE28 | TIE27 | TIE26 | TIE25 | TIE24 |
| 0xE4 | TXBCIE | 7:0 | CFIE7 | CFIE6 | CFIE5 | CFIE4 | CFIE3 | CFIE2 | CFIE1 | CFIE0 |
| | | 15:8 | CFIE15 | CFIE14 | CFIE13 | CFIE12 | CFIE11 | CFIE10 | CFIE9 | CFIE8 |
| | | 23:16 | CFIE23 | CFIE22 | CFIE21 | CFIE20 | CFIE19 | CFIE18 | CFIE17 | CFIE16 |
| | | 31:24 | CFIE31 | CFIE30 | CFIE29 | CFIE28 | CFIE27 | CFIE26 | CFIE25 | CFIE24 |
| 0xE8 ... 0xEF | Reserved | | | | | | | | | |
| 0xF0 | TXEFC | 7:0 | EFSA[7:0] | | | | | | | |
| | | 15:8 | EFSA[15:8] | | | | | | | |
| | | 23:16 | | | EFS[5:0] | | | | | |
| | | 31:24 | | | EFWM[5:0] | | | | | |
| 0xF4 | TXEFS | 7:0 | | | EFFL[5:0] | | | | | |
| | | 15:8 | | | | EFGI[4:0] | | | | |
| | | 23:16 | | | | EFPI[4:0] | | | | |
| | | 31:24 | | | | | | | TEFL | EFF |
| 0xF8 | TXEFA | 7:0 | | | | EFAI[4:0] | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 39.8. Register Description

Registers are 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.1. Core Release

**Name:** CREL
**Offset:** 0x00
**Reset:** 0x32100000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | REL[3:0] | | | | STEP[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SUBSTEP[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 1 | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bits 31:28 – REL[3:0]** Core Release
One digit, BCD-coded.

**Bits 27:24 – STEP[3:0]** Step of Core Release
One digit, BCD-coded.

**Bits 23:20 – SUBSTEP[3:0]** Sub-step of Core Release
One digit, BCD-coded.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.2. Endian

**Name:** ENDN
**Offset:** 0x04
**Reset:** 0x87654321
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ETV[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ETV[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ETV[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

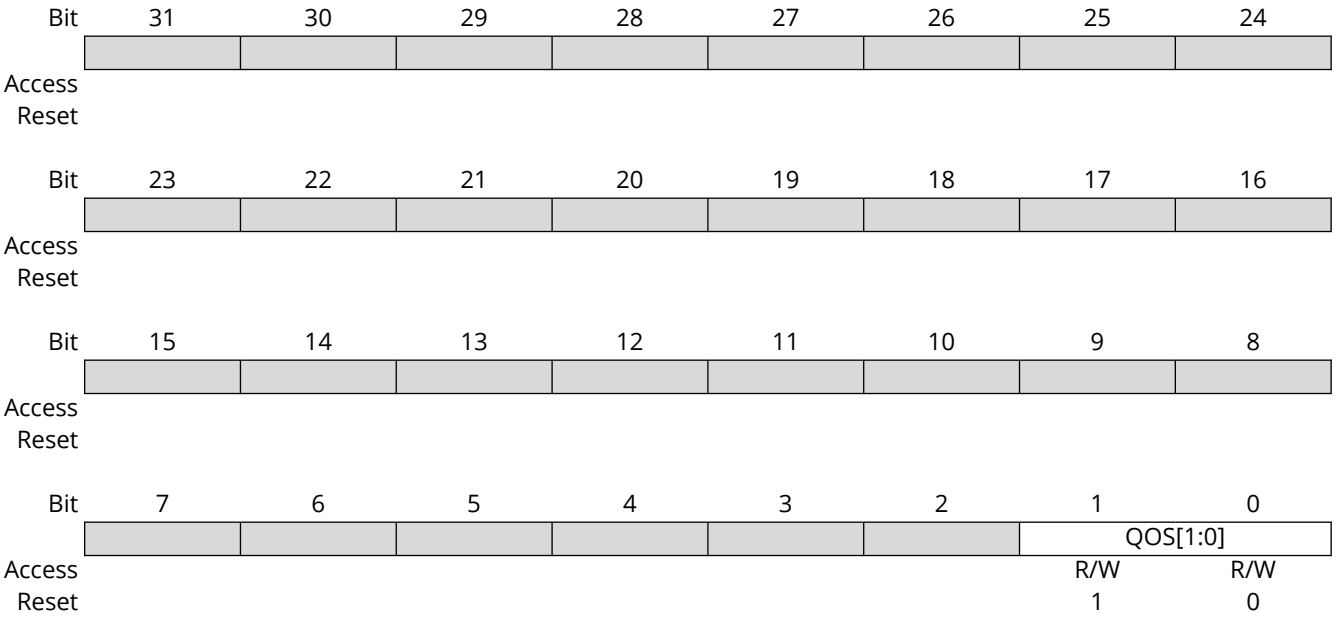
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ETV[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

**Bits 31:0 – ETV[31:0]** Endianness Test Value
The endianness test value is 0x87654321


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.3. Message RAM Configuration

**Name:** MRCFG
**Offset:** 0x08
**Reset:** 0x00000002
**Property:** -




**Bits 1:0 – QOS[1:0]** Data Quality of Service
This field defines the memory priority access during the Message RAM read/write data operation.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Background (no sensitive operation) |
| 0x1 | LOW | Sensitive bandwidth |
| 0x2 | MEDIUM | Sensitive latency |
| 0x3 | HIGH | Critical latency |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.4. Data Bit Timing and Prescaler

**Name:** DBTP
**Offset:** 0x0C
**Reset:** 0x00000A33
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

The CAN bit time may be programmed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 GCLK_CAN periods. $t_q$ = (DBRP + 1) mtq.

**Note:**
With a GCLK_CAN of 8MHz, the reset value 0x00000A33 configures the CAN for a fast bit rate of 500 kBits/s.

The bit rate configured for the CAN FD data phase via DBTP must be higher or equal to the bit rate configured for the arbitration phase via NBTP.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TDC | | | DBRP[4:0] | | | | |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | DTSEG1[4:0] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 1 | 0 | 1 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DTSEG2[3:0] | | | | DSJW[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

**Bit 23 – TDC** Transceiver Delay Compensation

| Value | Description |
|---|---|
| 0 | Transceiver Delay Compensation disabled. |
| 1 | Transceiver Delay Compensation enabled. |

**Bits 20:16 – DBRP[4:0]** Data Baud Rate Prescaler

| Value | Description |
|---|---|
| 0x00 – 0x1F | The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. |

**Bits 12:8 – DTSEG1[4:0]** Fast time segment before sample point

| Value | Description |
|---|---|
| 0x00 – 0x1F | Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG1 is the sum of Prop_Seg and Phase_Seg1. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 7:4 – DTSEG2[3:0]** Data time segment after sample point

| Value | Description |
|---|---|
| 0x0 - 0xF | Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. DTSEG2 is Phase_Seg2. |

**Bits 3:0 – DSJW[3:0]** Data (Re)Syncronization Jump Width

| Value | Description |
|---|---|
| 0x0 - 0xF | Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 39.8.5. Test

**Name:** TEST
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** Read-only, Write-restricted

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RX | TX[1:0] | | LBCK | | | | |
| Access | R | R/W | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

**Bit 7 – RX** Receive Pin
Monitors the actual value of pin CAN_RX

| Value | Description |
|---|---|
| 0 | The CAN bus is dominant (CAN_RX = 0). |
| 1 | The CAN bus is recessive (CAN_RX = 1). |

**Bits 6:5 – TX[1:0]** Control of Transmit Pin
This field defines the control of the transmit pin.

| Value | Name | Description |
|---|---|---|
| 0x0 | CORE | Reset value, CAN_TX controlled by CAN core, updated at the end of CAN bit time. |
| 0x1 | SAMPLE | Sample Point can be monitored at pin CAN_TX. |
| 0x2 | DOMINANT | Dominant (‘0’) level at pin CAN_TX. |
| 0x3 | RECESSIVE | Recessive (‘1’) level at pin CAN_TX. |

**Bit 4 – LBCK** Loop Back Mode

| Value | Description |
|---|---|
| 0 | Loop Back Mode is disabled. |
| 1 | Loop Back Mode is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.6. RAM Watchdog

**Name:** RWD
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** Read-only, Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the CAN's AHB Host Interface starts the Message RAM Watchdog Counter with the value configured by RWD.WDC. The counter is reloaded with RWD.WDC when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt IR.WDI is set.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WDV[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WDC[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:8 – WDV[7:0]** Watchdog Value
Actual Message RAM Watchdog Counter Value.

**Bits 7:0 – WDC[7:0]** Watchdog Configuration
Start value of the Message RAM Watchdog Counter. With the reset value of 0x00 the counter is disabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.7. CC Control

**Name:** CCCR
**Offset:** 0x18
**Reset:** 0x00000001
**Property:** Read-only, Write-restricted

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | TXP | EFBI | PXHD | | | BRSE | FDOE |
| Access | | R/W | R/W | R/W | | | R/W | R/W |
| Reset | | 0 | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TEST | DAR | MON | CSR | CSA | ASM | CCE | INIT |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bit 14 – TXP** Transmit Pause
This bit field is write-restricted and only writable if bit fields CCE = 1 and INIT = 1.

| Value | Description |
|---|---|
| 0 | Transmit pause disabled. |
| 1 | Transmit pause enabled. The CAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame. |

**Bit 13 – EFBI** Edge Filtering during Bus Integration

| Value | Description |
|---|---|
| 0 | Edge filtering is disabled. |
| 1 | Two consecutive dominant tq required to detect an edge for hard synchronization. |

**Bit 12 – PXHD** Protocol Exception Handling Disable
**Note:** When protocol exception handling is disabled, the CAN will transmit an error frame when it detects a protocol exception condition.

| Value | Description |
|---|---|
| 0 | Protocol exception handling enabled. |
| 1 | Protocol exception handling disabled. |

**Bit 9 – BRSE** Bit Rate Switch Enable
**Note:** When CAN FD operation is disabled FDOE = 0, BRSE is not evaluated.

| Value | Description |
|---|---|
| 0 | Bit rate switching for transmissions disabled. |
| 1 | Bit rate switching for transmissions enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 8 – FDOE** FD Operation Enable

| Value | Description |
|---|---|
| 0 | FD operation disabled. |
| 1 | FD operation enabled. |

**Bit 7 – TEST** Test Mode Enable

This bit field is write-restricted.
Writing a 0 to this field is always allowed.
Writing a 1 to this field is only allowed if bit fields CCE = 1 and INIT = 1.

| Value | Description |
|---|---|
| 0 | Normal operation. Register TEST holds reset values. |
| 1 | Test Mode, write access to register TEST enabled. |

**Bit 6 – DAR** Disable Automatic Retransmission

This bit field is write-restricted and only writable if bit fields CCE = 1 and INIT = 1.

| Value | Description |
|---|---|
| 0 | Automatic retransmission of messages not transmitted successfully enabled. |
| 1 | Automatic retransmission disabled. |

**Bit 5 – MON** Bus Monitoring Mode

This bit field is write-restricted.
Writing a 0 to this field is always allowed.
Writing a 1 to this field is only allowed if bit fields CCE = 1 and INIT = 1.

| Value | Description |
|---|---|
| 0 | Bus Monitoring Mode is disabled. |
| 1 | Bus Monitoring Mode is enabled. |

**Bit 4 – CSR** Clock Stop Request

| Value | Description |
|---|---|
| 0 | No clock stop is requested. |
| 1 | Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle. |

**Bit 3 – CSA** Clock Stop Acknowledge

| Value | Description |
|---|---|
| 0 | No clock stop acknowledged. |
| 1 | CAN may be set in power down by stopping CLK_CANx_AHB and GCLK_CAN. |

**Bit 2 – ASM** Restricted Operation Mode

This bit field is write-restricted.
Writing a 0 to this field is always allowed.
Writing a 1 to this field is only allowed if bit fields CCE = 1 and INIT = 1.

| Value | Description |
|---|---|
| 0 | Normal CAN operation. |
| 1 | Restricted Operation Mode active. |

**Bit 1 – CCE** Configuration Change Enable

This bit field is write-restricted and only writable if bit field INIT = 1.

| Value | Description |
|---|---|
| 0 | The CPU has no write access to the protected configuration registers. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The CPU has write access to the protected configuration registers (while CCCR.INIT = 1). |

**Bit 0 – INIT** Initialization

Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. The programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.

| Value | Description |
|---|---|
| 0 | Normal Operation. |
| 1 | Initialization is started. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.8. Nominal Bit Timing and Prescaler

**Name:** NBTP
**Offset:** 0x1C
**Reset:** 0x06000A03
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 GCLK_CAN periods. $t_q$ = (NBRP + 1) mtq.

NTSEG1 is the sum of Prop_Seg and Phase_Seg1. NTSEG2 is Phase_Seg2.

Therefore the length of the bit time is (programmed values) [NTSEG1 + NTSEG2 + 3] tq or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq.

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

**Note:** With a CAN clock (GCLK_CAN) of 8 MHz, the reset value 0x06000A03 configures the CAN for a bit rate of 500 Kbits.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | NSJW[6:0] | | | | | | | NBRP[8] |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | NBRP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NTSEG1[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | NTSEG2[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**Bits 31:25 – NSJW[6:0]** Nominal (Re)Syncronization Jump Width

| Value | Description |
|---|---|
| 0x00 – 0x7F | Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. |

**Bits 24:16 – NBRP[8:0]** Nominal Baud Rate Prescaler

| Value | Description |
|---|---|
| 0x000 – 0x1FF | The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. |

**Bits 15:8 – NTSEG1[7:0]** Nominal Time segment before sample point


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0x00 – 0x7F | Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG1 is the sum of Prop_Seg and Phase_Seg1. |

**Bits 6:0 – NTSEG2[6:0]** Time segment after sample point

| Value | Description |
|---|---|
| 0x00 – 0x7F | Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the programmed value is used. NTSEG2 is Phase_Seg2. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 39.8.9. Timestamp Counter Configuration

**Name:** TSCC
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | TCP[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | TSS[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 19:16 – TCP[3:0]** Timestamp Counter Prescaler

| Value | Description |
|---|---|
| 0x0 – 0xF | Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1...16]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. |

**Bits 1:0 – TSS[1:0]** Timestamp Select
This field defines the timestamp counter selection.

| Value | Name | Description |
|---|---|---|
| 0x0 or 0x3 | ZERO | Timestamp counter value always 0x0000. |
| 0x1 | INC | Timestamp counter value incremented by TCP. |
| 0x2 | Reserved | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.10. Timestamp Counter Value

**Name:** TSCV
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** Read-only

**Notes:**

1. A write access to TSCV while in internal mode clears the Timestamp Counter value. A write access to TSCV while in external mode has no impact.
2. A “wrap around” is a change of the Timestamp Counter value from non-zero to zero not caused by the write access to TSCV.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TSC[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TSC[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – TSC[15:0]** Timestamp Counter
The internal Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSCC.TSS = 0x1, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of TSCC.TCP. A wrap around sets interrupt flag IR.TSW.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.11. Timeout Counter Configuration

**Name:** TOCC
**Offset:** 0x28
**Reset:** 0xFFFF0000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TOP[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TOP[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | TOS[1:0] | | ETOC |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 31:16 – TOP[15:0]** Timeout Period
Start value of the Timeout Counter (down-counter). Configures the Timeout Period.

**Bits 2:1 – TOS[1:0]** Timeout Select
When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC.TOP and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC.TOP. Down-counting is started when the first FIFO element is stored.

| Value | Name | Description |
|---|---|---|
| 0x0 | CONT | Continuous operation. |
| 0x1 | TXEF | Timeout controlled by TX Event FIFO. |
| 0x2 | RXF0 | Timeout controlled by Rx FIFO 0. |
| 0x3 | RXF1 | Timeout controlled by Rx FIFO 1. |

**Bit 0 – ETOC** Enable Timeout Counter

| Value | Description |
|---|---|
| 0 | Timeout Counter disabled. |
| 1 | Timeout Counter enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.12. Timeout Counter Value

**Name:** TOCV
**Offset:** 0x2C
**Reset:** 0x0000FFFF
**Property:** Read-only

**Note:** A write access to TOCV reloads the Timeout Counter with the value of TOCV.TOP.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TOC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TOC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 15:0 – TOC[15:0]** Timeout Counter
The Timeout Counter is decremented in multiples of CAN bit times [1...16] depending on the configuration of TSCC.TCP. When decremented to zero, interrupt flag IR.TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC.TOS.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.13. Error Counter

**Name:** ECR
**Offset:** 0x40
**Reset:** 0x00000000
**Property:** Read-only

**Note:** When CCCR.ASM is set, the CAN protocol controller does not increment TECand REC when a CAN protocol error is detected, but CEL is still incremented.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CEL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RP | REC[6:0] | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TEC[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – CEL[7:0]** CAN Error Logging
The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR.ELO.

**Bit 15 – RP** Receive Error Passive

**Bits 14:8 – REC[6:0]** Receive Error Counter
Actual state of the Receive Error Counter, values between 0 and 127.

**Bits 7:0 – TEC[7:0]** Transmit Error Counter
Actual state of the Transmit Error Counter, values between 0 and 255.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.14. Protocol Status

**Name:** PSR
**Offset:** 0x44
**Reset:** 0x00000707
**Property:** Read-only

**Notes:**

1. When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in FLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.
2. The Bus_Off recovery sequence (see CAN Specification Rev. 2.0 or ISO 11898-1) cannot be shortened by setting or resetting CCCR.INIT. If the device goes Bus_Off, it will set CCCR.INIT of its own accord, stopping all bus activities. Once CCCR.INIT has been cleared by the CPU, the device will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits has been monitored, a Bit0 Error code is written to PSR.LEC, enabling the CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. ECR.REC is used to count these sequences.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | TDCV[6:0] | | | | | | |
| Access | | R | R | R | R | R | R | R |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | PXE | RFDF | RBRS | RESI | DLEC[2:0] | | |
| Access | | R | R | R | R | R | R | R |
| Reset | | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BO | EW | EP | ACT[1:0] | | LEC[2:0] | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

**Bits 22:16 – TDCV[6:0]** Transmitter Delay Compensation Value

| Value | Description |
|---|---|
| 0x00 – 0x7F | Position of the secondary sample point, defined by the sum of the measured delay from CAN_TX to CAN_RX and TDCR.TDCO. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq. |

**Bit 14 – PXE** Protocol Exception Event
This field is cleared on read access.

| Value | Description |
|---|---|
| 0 | No protocol exception event occurred since last read access. |
| 1 | Protocol exception event occurred. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 13 – RFDF** Received a CAN FD Message
This field is cleared on read access.

| Value | Description |
|---|---|
| 0 | Since this bit was reset by the CPU, no CAN FD message has been received. |
| 1 | Message in CAN FD format with FDF flag set has been received. This bit is set independent of acceptance filtering. |

**Bit 12 – RBRS** BRS flag of last received CAN FD Message
This field is cleared on read access.

| Value | Description |
|---|---|
| 0 | Last received CAN FD message did not have its BRS flag set. |
| 1 | Last received CAN FD message had its BRS flag set. This bit is set together with RFDF, independent of acceptance filtering. |

**Bit 11 – RESI** ESI flag of last received CAN FD Message
This field is cleared on read access.

| Value | Description |
|---|---|
| 0 | Last received CAN FD message did not have its ESI flag set. |
| 1 | Last received CAN FD message had its ESI flag set. |

**Bits 10:8 – DLEC[2:0]** Data Last Error Code
Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.

**Bit 7 – BO** Bus_Off Status

| Value | Description |
|---|---|
| 0 | The CAN is not Bus_Off. |
| 1 | The CAN is in Bus_Off state. |

**Bit 6 – EW** Error Warning Status

| Value | Description |
|---|---|
| 0 | Both error counters are below the Error_Warning limit of 96. |
| 1 | At least one of the error counter has reached the Error_Warning limit of 96. |

**Bit 5 – EP** Error Passive

| Value | Description |
|---|---|
| 0 | The CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected. |
| 1 | The CAN is in the Error_Passive state. |

**Bits 4:3 – ACT[1:0]** Activity
Monitors the module's CAN communication state.

| Value | Name | Description |
|---|---|---|
| 0x0 | SYNC | Node is synchronizing on CAN communication. |
| 0x1 | IDLE | Node is neither receiver nor transmitter. |
| 0x2 | RX | Node is operating as receiver. |
| 0x3 | TX | Node is operating as transmitter. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 2:0 – LEC[2:0]** Last Error Code

The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error.
This field is set on read access.

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No Error: No error occurred since LEC has been reset by successful reception or transmission. |
| 0x1 | STUFF | Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. |
| 0x2 | FORM | Form Error: A fixed format part of a received frame has the wrong format. |
| 0x3 | ACK | Ack Error: The message transmitted by the CAN was not acknowledged by another node. |
| 0x4 | BIT1 | Bit1 Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus was dominant. |
| 0x5 | BIT0 | Bit0 Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits have been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). |
| 0x6 | CRC | CRC Error: The CRC checksum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data. |
| 0x7 | NC | No Change: Any read access to the Protocol Status Register re-initializes the LEC to '7'. When the LEC shows the value '7', no CAN bus event was detected since the last CPU read access to the Protocol Status Register. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.15. Transmitter Delay Compensation

**Name:** TDCR
**Offset:** 0x48
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | TDCO[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | TDCF[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 14:8 – TDCO[6:0]** Transmitter Delay Compensation Offset

| Value | Description |
|---|---|
| 0x00 – 0x7F | Offset value defining the distance between the measured delay from CAN_TX to CAN_RX and the secondary sample point. Valid values are 0 to 127 mtq. |

**Bits 6:0 – TDCF[6:0]** Transmitter Delay Compensation Filter Window Length

| Value | Description |
|---|---|
| 0x00 – 0x7F | Defines the minimum value for the SSP position, dominant edges on CAN_RX that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.16. Interrupt

**Name:** IR
**Offset:** 0x50
**Reset:** 0x00000000
**Property:** -

The flags are set when one of the listed conditions is detected (edge-sensitive). A flag is cleared by writing a 1 to the corresponding bit field. Writing a 0 has no effect. A hard reset will clear the register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | ARA | PED | PEA | WDI | BO | EW |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | EP | ELO | BEU | BEC | DRX | TOO | MRAF | TSW |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TEFL | TEFF | TEFW | TEFN | TFE | TCF | TC | HPM |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RF1L | RF1F | RF1W | RF1N | RF0L | RF0F | RF0W | RF0N |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 29 – ARA** Access to Reserved Address

| Value | Description |
|---|---|
| 0 | No access to reserved address occurred. |
| 1 | Access to reserved address occurred. |

**Bit 28 – PED** Protocol Error in Data Phase

| Value | Description |
|---|---|
| 0 | No protocol error in data phase. |
| 1 | Protocol error in data phase detected (PSR.DLEC != 0,7). |

**Bit 27 – PEA** Protocol Error in Arbitration Phase

| Value | Description |
|---|---|
| 0 | No protocol error in arbitration phase. |
| 1 | Protocol error in arbitration phase detected (PSR.LEC != 0,7). |

**Bit 26 – WDI** Watchdog Interrupt

| Value | Description |
|---|---|
| 0 | No Message RAM Watchdog event occurred. |
| 1 | Message RAM Watchdog event due to missing READY. |

**Bit 25 – BO** Bus_Off Status


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Bus_Off status unchanged. |
| 1 | Bus_Off status changed. |

**Bit 24 – EW** Error Warning Status

| Value | Description |
|---|---|
| 0 | Error_Warning status unchanged. |
| 1 | Error_Warning status changed. |

**Bit 23 – EP** Error Passive

| Value | Description |
|---|---|
| 0 | Error_Passive status unchanged. |
| 1 | Error_Passive status changed. |

**Bit 22 – ELO** Error Logging Overflow

| Value | Description |
|---|---|
| 0 | CAN Error Logging Counter did not overflow. |
| 1 | Overflow of CAN Error Logging Counter occurred. |

**Bit 21 – BEU** Bit Error Uncorrected

Message RAM bit error detected, uncorrected. Generated by an optional external parity / ECC logic attached to the Message RAM. An uncorrected Message RAM bit sets CCCR.INIT to 1. This is done to avoid transmission of corrupted data.

| Value | Description |
|---|---|
| 0 | Not bit error detected when reading from Message RAM. |
| 1 | Bit error detected, uncorrected (e.g. parity logic). |

**Bit 20 – BEC** Bit Error Corrected

Message RAM bit error detected and corrected. Generated by an optional external parity / ECC logic attached to the Message RAM.

| Value | Description |
|---|---|
| 0 | Not bit error detected when reading from Message RAM. |
| 1 | Bit error detected and corrected (e.g. ECC). |

**Bit 19 – DRX** Message stored to Dedicated Rx Buffer

The flag is set whenever a received message has been stored into a dedicated Rx Buffer.

| Value | Description |
|---|---|
| 0 | No Rx Buffer updated. |
| 1 | At least one received message stored into a Rx Buffer. |

**Bit 18 – TOO** Timeout Occurred

| Value | Description |
|---|---|
| 0 | No timeout. |
| 1 | Timeout reached. |

**Bit 17 – MRAF** Message RAM Access Failure

The flag is set, when the Rx Handler

- has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- was not able to write a message to the Message RAM. In this case message storage is aborted.

In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the CAN is switched into Restricted Operation Mode. To leave Restricted Operation Mode, the Host CPU has to reset CCCR.ASM.

| Value | Description |
|---|---|
| 0 | No Message RAM access failure occurred. |
| 1 | Message RAM access failure occurred. |

**Bit 16 – TSW** Timestamp Wraparound

| Value | Description |
|---|---|
| 0 | No timestamp counter wrap-around. |
| 1 | Timestamp counter wrapped around. |

**Bit 15 – TEFL** Tx Event FIFO Element Lost

| Value | Description |
|---|---|
| 0 | No Tx Event FIFO element lost. |
| 1 | Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero. |

**Bit 14 – TEFF** Tx Event FIFO Full

| Value | Description |
|---|---|
| 0 | Tx Event FIFO not full. |
| 1 | Tx Event FIFO full. |

**Bit 13 – TEFW** Tx Event FIFO Watermark Reached

| Value | Description |
|---|---|
| 0 | Tx Event FIFO fill level below watermark. |
| 1 | Tx Event FIFO fill level reached watermark. |

**Bit 12 – TEFN** Tx Event FIFO New Entry

| Value | Description |
|---|---|
| 0 | Tx Event FIFO unchanged. |
| 1 | Tx Handler wrote Tx Event FIFO element. |

**Bit 11 – TFE** Tx FIFO Empty

| Value | Description |
|---|---|
| 0 | Tx FIFO non-empty. |
| 1 | Tx FIFO empty. |

**Bit 10 – TCF** Transmission Cancellation Finished

| Value | Description |
|---|---|
| 0 | No transmission cancellation finished. |
| 1 | Transmission cancellation finished. |

**Bit 9 – TC** Timestamp Completed

| Value | Description |
|---|---|
| 0 | No transmission completed. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Transmission completed. |

**Bit 8 – HPM** High Priority Message

| Value | Description |
|---|---|
| 0 | No high priority message received. |
| 1 | High priority message received. |

**Bit 7 – RF1L** Rx FIFO 1 Message Lost

| Value | Description |
|---|---|
| 0 | No Rx FIFO 1 message lost. |
| 1 | Rx FIFO 1 message lost. also set after write attempt to Rx FIFO 1 of size zero. |

**Bit 6 – RF1F** Rx FIFO 1 Full

| Value | Description |
|---|---|
| 0 | Rx FIFO 1 not full. |
| 1 | Rx FIFO 1 full. |

**Bit 5 – RF1W** Rx FIFO 1 Watermark Reached

| Value | Description |
|---|---|
| 0 | Rx FIFO 1 fill level below watermark. |
| 1 | Rx FIFO 1 fill level reached watermark. |

**Bit 4 – RF1N** Rx FIFO 1 New Message

| Value | Description |
|---|---|
| 0 | No new message written to Rx FIFO 1. |
| 1 | New message written to Rx FIFO 1. |

**Bit 3 – RF0L** Rx FIFO 0 Message Lost

| Value | Description |
|---|---|
| 0 | No Rx FIFO 0 message lost. |
| 1 | Rx FIFO 0 message lost. also set after write attempt to Rx FIFO 0 of size zero. |

**Bit 2 – RF0F** Rx FIFO 0 Full

| Value | Description |
|---|---|
| 0 | Rx FIFO 0 not full. |
| 1 | Rx FIFO 0 full. |

**Bit 1 – RF0W** Rx FIFO 0 Watermark Reached

| Value | Description |
|---|---|
| 0 | Rx FIFO 0 fill level below watermark. |
| 1 | Rx FIFO 0 fill level reached watermark. |

**Bit 0 – RF0N** Rx FIFO 0 New Message

| Value | Description |
|---|---|
| 0 | No new message written to Rx FIFO 0. |
| 1 | New message written to Rx FIFO 0. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.17. Interrupt Enable

**Name:** IE
**Offset:** 0x54
**Reset:** 0x00000000
**Property:** -

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register will be signalled on an interrupt line.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | ARAE | PEDE | PEAE | WDIE | BOE | EWE |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | EPE | ELOE | BEUE | BECE | DRXE | TOOE | MRAFE | TSWE |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TEFLE | TEFFE | TEFWE | TEFNE | TFEE | TCFE | TCE | HPME |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RF1LE | RF1FE | RF1WE | RF1NE | RF0LE | RF0FE | RF0WE | RF0NE |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 29 – ARAE** Access to Reserved Address Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 28 – PEDE** Protocol Error in Data Phase Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 27 – PEAE** Protocol Error in Arbitration Phase Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 26 – WDIE** Watchdog Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 25 – BOE** Bus_Off Status Interrupt Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 24 – EWE** Error Warning Status Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 23 – EPE** Error Passive Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 22 – ELOE** Error Logging Overflow Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 21 – BEUE** Bit Error Uncorrected Interrupt Enable.

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 20 – BECE** Bit Error Corrected Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 19 – DRXE** Message stored to Dedicated Rx Buffer Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 18 – TOOE** Timeout Occurred Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 17 – MRAFE** Message RAM Access Failure Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 16 – TSWE** Timestamp Wraparound Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 15 – TEFLE** Tx Event FIFO Event Lost Interrupt Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 14 – TEFFE** Tx Event FIFO Full Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 13 – TEFWE** Tx Event FIFO Watermark Reached Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 12 – TEFNE** Tx Event FIFO New Entry Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 11 – TFEE** Tx FIFO Empty Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 10 – TCFE** Transmission Cancellation Finished Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 9 – TCE** Transmission Completed Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 8 – HPME** High Priority Message Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 7 – RF1LE** Rx FIFO 1 Message Lost Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 6 – RF1FE** Rx FIFO 1 Full Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 5 – RF1WE** Rx FIFO 1 Watermark Reached Interrupt Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 4 – RF1NE** Rx FIFO 1 New Message Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 3 – RF0LE** Rx FIFO 0 Message Lost Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 2 – RF0FE** Rx FIFO 0 Full Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 1 – RF0WE** Rx FIFO 0 Watermark Reached Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 0 – RF0NE** Rx FIFO 0 New Message Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.18. Interrupt Line Select

**Name:** ILS
**Offset:** 0x58
**Reset:** 0x00000000
**Property:** -

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from IR to one of the two module interrupt lines.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | ARAL | PEDL | PEAL | WDIL | BOL | EWL |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | EPL | ELOL | BEUL | BECL | DRXL | TOOL | MRAFL | TSWL |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TEFLL | TEFFL | TEFWL | TEFNL | TFEL | TCFL | TCL | HPML |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RF1LL | RF1FL | RF1WL | RF1NL | RF0LL | RF0FL | RF0WL | RF0NL |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 29 – ARAL** Access to Reserved Address Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 28 – PEDL** Protocol Error in Data Phase Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 27 – PEAL** Protocol Error in Arbitration Phase Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 26 – WDIL** Watchdog Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 25 – BOL** Bus_Off Status Interrupt Line


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 24 – EWL** Error Warning Status Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 23 – EPL** Error Passive Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 22 – ELOL** Error Logging Overflow Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 21 – BEUL** Bit Error Uncorrected Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 20 – BECL** Bit Error Corrected Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 19 – DRXL** Message stored to Dedicated Rx Buffer Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 18 – TOOL** Timeout Occurred Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 17 – MRAFL** Message RAM Access Failure Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 16 – TSWL** Timestamp Wraparound Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 15 – TEFLL** Tx Event FIFO Event Lost Interrupt Line


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 14 – TEFFL** Tx Event FIFO Full Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 13 – TEFWL** Tx Event FIFO Watermark Reached Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 12 – TEFNL** Tx Event FIFO New Entry Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 11 – TFEL** Tx FIFO Empty Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 10 – TCFL** Transmission Cancellation Finished Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 9 – TCL** Transmission Completed Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 8 – HPML** High Priority Message Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 7 – RF1LL** Rx FIFO 1 Message Lost Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 6 – RF1FL** Rx FIFO 1 Full Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 5 – RF1WL** Rx FIFO 1 Watermark Reached Interrupt Line


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 4 – RF1NL** Rx FIFO 1 New Message Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 3 – RF0LL** Rx FIFO 0 Message Lost Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 2 – RF0FL** Rx FIFO 0 Full Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 1 – RF0WL** Rx FIFO 0 Watermark Reached Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 0 – RF0NL** Rx FIFO 0 New Message Interrupt Line

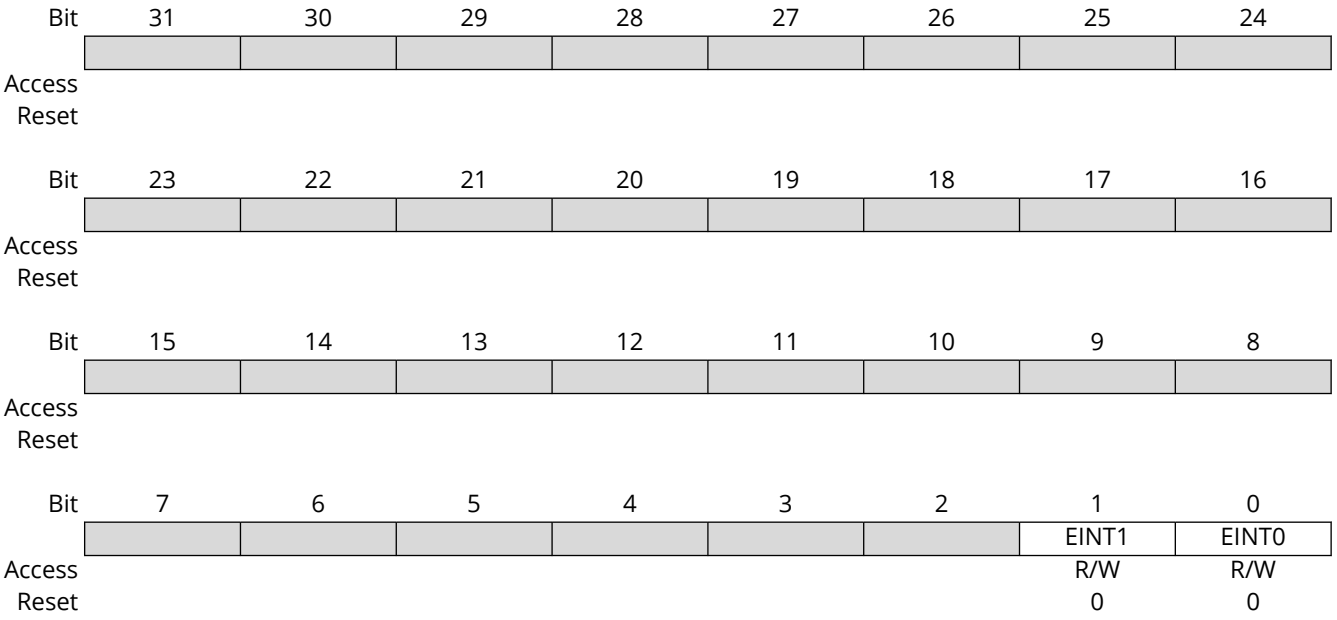
| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 39.8.19. Interrupt Line Enable

**Name:** ILE
**Offset:** 0x5C
**Reset:** 0x00000000
**Property:** -




**Bits 0, 1 – EINTn** Enable Interrupt Line n [n = 1,0]

| Value | Description |
|---|---|
| 0 | CAN interrupt line n disabled. |
| 1 | CAN interrupt line n enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.20. Global Filter Configuration

**Name:** GFC
**Offset:** 0x80
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | ANFS[1:0] | | ANFE[1:0] | | RRFS | RRFE |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:4 – ANFS[1:0]** Accept Non-matching Frames Standard
Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.

| Value | Name | Description |
|---|---|---|
| 0x0 | RXF0 | Accept in Rx FIFO 0. |
| 0x1 | RXF1 | Accept in Rx FIFO 1. |
| 0x2 or 0x3 | REJECT | Reject |

**Bits 3:2 – ANFE[1:0]** Accept Non-matching Frames Extended
Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.

| Value | Name | Description |
|---|---|---|
| 0x0 | RXF0 | Accept in Rx FIFO 0. |
| 0x1 | RXF1 | Accept in Rx FIFO 1. |
| 0x2 or 0x3 | REJECT | Reject |

**Bit 1 – RRFS** Reject Remote Frames Standard

| Value | Description |
|---|---|
| 0 | Filter remote frames with 11-bit standard IDs. |
| 1 | Reject all remote frames with 11-bit standard IDs. |

**Bit 0 – RRFE** Reject Remote Frames Extended


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Filter remote frames with 29-bit extended IDs. |
| 1 | Reject all remote frames with 29-bit extended IDS. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 39.8.21. Standard ID Filter Configuration

**Name:** SIDFC
**Offset:** 0x84
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | LSS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FLSSA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FLSSA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – LSS[7:0]** List Size Standard

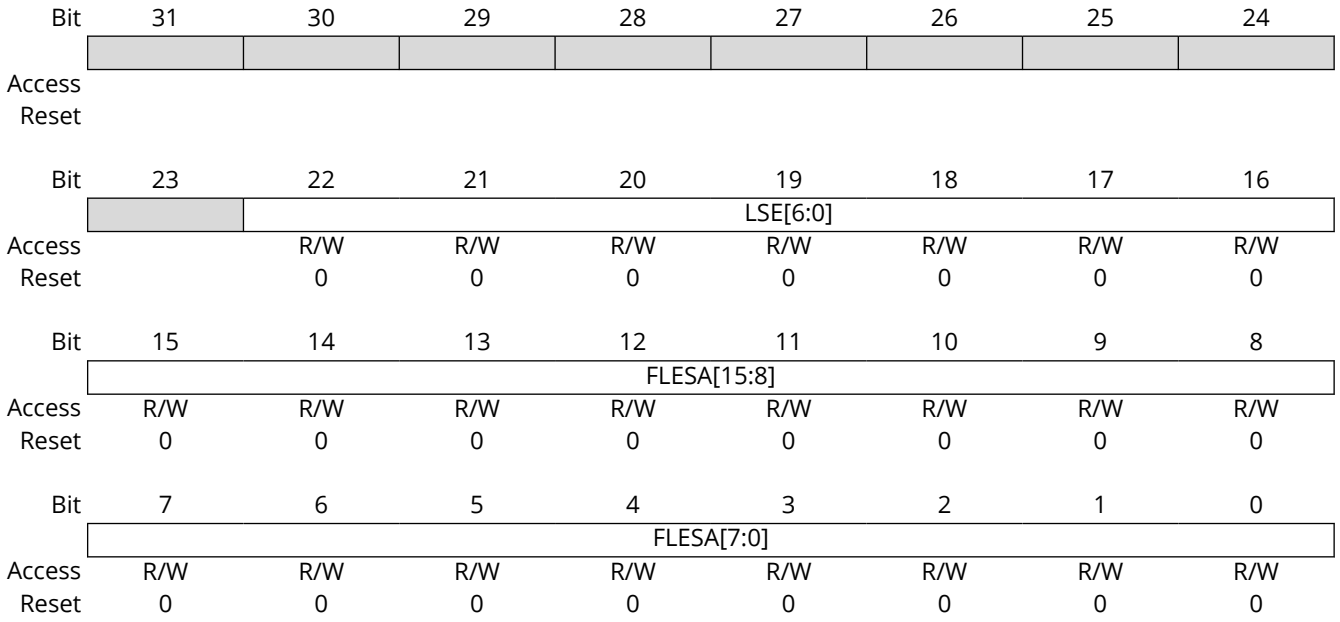
| Value | Description |
|---|---|
| 0 | No standard Message ID filter. |
| 1 - 128 | Number of standard Message ID filter elements. |
| > 128 | Values greater than 128 are interpreted as 128. |

**Bits 15:0 – FLSSA[15:0]** Filter List Standard Start Address
Start address of standard Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.22. Extended ID Filter Configuration

**Name:** XIDFC
**Offset:** 0x88
**Reset:** 0x00000000
**Property:** Write-restricted



| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | LSE[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FLESA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FLESA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 22:16 – LSE[6:0]** List Size Extended

| Value | Description |
|---|---|
| 0 | No extended Message ID filter. |
| 1 – 64 | Number of Extended Message ID filter elements. |
| > 64 | Values greater than 64 are interpreted as 64. |

**Bits 15:0 – FLESA[15:0]** Filter List Extended Start Address
Start address of extended Message ID filter list. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.23. Extended ID AND Mask

**Name:** XIDAM
**Offset:** 0x90
**Reset:** 0x1FFFFFFF
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | EIDM[28:24] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 1 | 1 | 1 | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | EIDM[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EIDM[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EIDM[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 28:0 – EIDM[28:0]** Extended ID Mask
For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.24. High Priority Message Status

**Name:** HPMS
**Offset:** 0x94
**Reset:** 0x00000000
**Property:** Read-only

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FLST | FIDX[6:0] | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MSI[1:0] | | BIDX[5:0] | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – FLST** Filter List
Indicates the filter list of the matching filter element.

| Value | Description |
|---|---|
| 0 | Standard Filter List. |
| 1 | Extended Filter List. |

**Bits 14:8 – FIDX[6:0]** Filter Index
Index of matching filter element. Range is 0 to SIDFC.LSS - 1 (standard) or XIDFC.LSE - 1 (extended).

**Bits 7:6 – MSI[1:0]** Message Storage Indicator
This field defines the message storage information to a FIFO.

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No FIFO selected. |
| 0x1 | LOST | FIFO message lost. |
| 0x2 | FIFO0 | Message stored in FIFO 0. |
| 0x3 | FIFO1 | Message stored in FIFO 1. |

**Bits 5:0 – BIDX[5:0]** Buffer Index
Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = 1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.25. New Data 1

**Name:** NDAT1
**Offset:** 0x98
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ND31 | ND30 | ND29 | ND28 | ND27 | ND26 | ND25 | ND24 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ND23 | ND22 | ND21 | ND20 | ND19 | ND18 | ND17 | ND16 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ND15 | ND14 | ND13 | ND12 | ND11 | ND10 | ND9 | ND8 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ND7 | ND6 | ND5 | ND4 | ND3 | ND2 | ND1 | ND0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NDn** New Data n [n = 0..31]

The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.26. New Data 2

**Name:** NDAT2
**Offset:** 0x9C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ND63 | ND62 | ND61 | ND60 | ND59 | ND58 | ND57 | ND56 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ND55 | ND54 | ND53 | ND52 | ND51 | ND50 | ND49 | ND48 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ND47 | ND46 | ND45 | ND44 | ND43 | ND42 | ND41 | ND40 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ND39 | ND38 | ND37 | ND36 | ND35 | ND34 | ND33 | ND32 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NDn** New Data [n = 32..64]

The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host clears them. A flag is cleared by writing 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.27. Rx FIFO 0 Configuration

**Name:** RXF0C
**Offset:** 0xA0
**Reset:** 0x00000000
**Property:** Write-restricted

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | F0OM | F0WM[6:0] | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | F0S[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | F0SA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | F0SA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – F0OM** FIFO 0 Operation Mode
FIFO 0 can be operated in blocking or in overwrite mode.

| Value | Description |
|---|---|
| 0 | FIFO 0 blocking mode. |
| 1 | FIFO 0 overwrite mode. |

**Bits 30:24 – F0WM[6:0]** Rx FIFO 0 Watermark

| Value | Description |
|---|---|
| 0 | Watermark interrupt disabled. |
| 1 – 64 | Level for Rx FIFO 0 watermark interrupt (IR.RF0W). |
| >64 | Watermark interrupt disabled. |

**Bits 22:16 – F0S[6:0]** Rx FIFO 0 Size
The Rx FIFO 0 elements are indexed from 0 to F0S - 1.

| Value | Description |
|---|---|
| 0 | No Rx FIFO 0 |
| 1 – 64 | Number of Rx FIFO 0 elements. |
| >64 | Values greater than 64 are interpreted as 64. |

**Bits 15:0 – F0SA[15:0]** Rx FIFO 0 Start Address
Start address of Rx FIFO 0 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.28. Rx FIFO 0 Status

**Name:** RXF0S
**Offset:** 0xA4
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RF0L | F0F |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | F0PI[5:0] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | F0GI[5:0] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | F0FL[6:0] | | | | | | |
| Access | | R | R | R | R | R | R | R |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 25 – RF0L** Rx FIFO 0 Message Lost
This bit is a copy of interrupt flag IR.RF0L. When IR.RF0L is reset, this bit is also reset.
Overwriting the oldest message when RXF0C.F0OM = '1' will not set this flag.

| Value | Description |
|---|---|
| 0 | No Rx FIFO 0 message lost. |
| 1 | Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero. |

**Bit 24 – F0F** Rx FIFO 0 Full

| Value | Description |
|---|---|
| 0 | Rx FIFO 0 not full. |
| 1 | Rx FIFO 0 full. |

**Bits 21:16 – F0PI[5:0]** Rx FIFO 0 Put Index
Rx FIFO 0 write index pointer, range 0 to 63.

**Bits 13:8 – F0GI[5:0]** Rx FIFO 0 Get Index
Rx FIFO 0 read index pointer, range 0 to 63.

**Bits 6:0 – F0FL[6:0]** Rx FIFO 0 Fill Level
Number of elements stored in Rx FIFO 0, range 0 to 64.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.29. Rx FIFO 0 Acknowledge

**Name:** RXF0A
**Offset:** 0xA8
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | F0AI[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:0 – F0AI[5:0]** Rx FIFO 0 Acknowledge Index
After the Host has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S.F0GI to F0AI + 1 and update the FIFO 0 Fill Level RXF0S.F0FL.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.30. Rx Buffer Configuration

**Name:** RXBC
**Offset:** 0xAC
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RBSA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RBSA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RBSA[15:0]** Rx Buffer Start Address
Configures the start address of the Rx Buffers section in the Message RAM. Also used to reference debug message A,B,C. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.31. Rx FIFO 1 Configuration

**Name:** RXF1C
**Offset:** 0xB0
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | F1OM | F1WM[6:0] | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | F1S[6:0] | | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | F1SA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | F1SA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – F1OM** FIFO 1 Operation Mode
FIFO 1 can be operated in blocking or in overwrite mode.

| Value | Description |
|---|---|
| 0 | FIFO 1 blocking mode. |
| 1 | FIFO 1 overwrite mode. |

**Bits 30:24 – F1WM[6:0]** Rx FIFO 1 Watermark

| Value | Description |
|---|---|
| 0 | Watermark interrupt disabled. |
| 1 – 64 | Level for Rx FIFO 1 watermark interrupt (IR.RF1W). |
| >64 | Watermark interrupt disabled. |

**Bits 22:16 – F1S[6:0]** Rx FIFO 1 Size
The Rx FIFO 1 elements are indexed from 0 to F1S - 1.

| Value | Description |
|---|---|
| 0 | No Rx FIFO 1 |
| 1 – 64 | Number of Rx FIFO 1 elements. |
| >64 | Values greater than 64 are interpreted as 64. |

**Bits 15:0 – F1SA[15:0]** Rx FIFO 1 Start Address
Start address of Rx FIFO 1 in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.32. Rx FIFO 1 Status

**Name:** RXF1S
**Offset:** 0xB4
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DMS[1:0] | | | | | | RF1L | F1F |
| Access | R | R | | | | | R | R |
| Reset | 0 | 0 | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | F1PI[5:0] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | F1GI[5:0] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | F1FL[6:0] | | | | | | |
| Access | | R | R | R | R | R | R | R |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:30 – DMS[1:0]** Debug Message Status
This field defines the debug message status.

| Value | Name | Description |
|---|---|---|
| 0x0 | IDLE | Idle state, wait for reception of debug messages, DMA request is cleared. |
| 0x1 | DBGA | Debug message A received. |
| 0x2 | DBGB | Debug message A, B received. |
| 0x3 | DBGC | Debug message A, B, C received, DMA request is set. |

**Bit 25 – RF1L** Rx FIFO 1 Message Lost
This bit is a copy of interrupt flag IR.RF1L. When IR.RF1L is reset, this bit is also reset.
Overwriting the oldest message when RXF1C.F1OM = '1' will not set this flag.

| Value | Description |
|---|---|
| 0 | No Rx FIFO 1 message lost. |
| 1 | Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero. |

**Bit 24 – F1F** Rx FIFO 1 Full

| Value | Description |
|---|---|
| 0 | Rx FIFO 1 not full. |
| 1 | Rx FIFO 1 full. |

**Bits 21:16 – F1PI[5:0]** Rx FIFO 1 Put Index
Rx FIFO 1 write index pointer, range 0 to 63.

**Bits 13:8 – F1GI[5:0]** Rx FIFO 1 Get Index
Rx FIFO 1 read index pointer, range 0 to 63.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 6:0 – F1FL[6:0]** Rx FIFO 1 Fill Level
Number of elements stored in Rx FIFO 1, range 0 to 64.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 39.8.33. Rx FIFO 1 Acknowledge

**Name:** RXF1A
**Offset:** 0xB8
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | F1AI[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:0 – F1AI[5:0]** Rx FIFO 1 Acknowledge Index
After the Host has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S.F1GI to F1AI + 1 and update the FIFO 1 Fill Level RXF1S.F1FL.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.34. Rx Buffer / FIFO Element Size Configuration

**Name:** RXESC
**Offset:** 0xBC
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | RBDS[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | F1DS[2:0] | | | | F0DS[2:0] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 10:8 – RBDS[2:0]** Rx Buffer Data Field Size
In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer, only the number of bytes as configured by RXESC are stored to the Rx Buffer element. The rest of the frame's data field is ignored.

| Value | Name | Description |
|---|---|---|
| 0x0 | DATA8 | 8 byte data field. |
| 0x1 | DATA12 | 12 byte data field. |
| 0x2 | DATA16 | 16 byte data field. |
| 0x3 | DATA20 | 20 byte data field. |
| 0x4 | DATA24 | 24 byte data field. |
| 0x5 | DATA32 | 32 byte data field. |
| 0x6 | DATA48 | 48 byte data field. |
| 0x7 | DATA64 | 64 byte data field. |

**Bits 6:4 – F1DS[2:0]** Rx FIFO 1 Data Field Size
In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 1, only the number of bytes as configured by RXESC are stored to the Rx FIFO 1 element. The rest of the frame's data field is ignored.

| Value | Name | Description |
|---|---|---|
| 0x0 | DATA8 | 8 byte data field. |
| 0x1 | DATA12 | 12 byte data field. |
| 0x2 | DATA16 | 16 byte data field. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x3 | DATA20 | 20 byte data field. |
| 0x4 | DATA24 | 24 byte data field. |
| 0x5 | DATA32 | 32 byte data field. |
| 0x6 | DATA48 | 48 byte data field. |
| 0x7 | DATA64 | 64 byte data field. |

**Bits 2:0 – F0DS[2:0]** Rx FIFO 0 Data Field Size
In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx FIFO 0, only the number of bytes as configured by RXESC are stored to the Rx FIFO 0 element. The rest of the frame's data field is ignored.

| Value | Name | Description |
|---|---|---|
| 0x0 | DATA8 | 8 byte data field. |
| 0x1 | DATA12 | 12 byte data field. |
| 0x2 | DATA16 | 16 byte data field. |
| 0x3 | DATA20 | 20 byte data field. |
| 0x4 | DATA24 | 24 byte data field. |
| 0x5 | DATA32 | 32 byte data field. |
| 0x6 | DATA48 | 48 byte data field. |
| 0x7 | DATA64 | 64 byte data field. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.35. Tx Buffer Configuration

**Name:** TXBC
**Offset:** 0xC0
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

**Note:** Be aware that the sum of TFQS and NDTB may not be greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | TFQM | TFQS[5:0] | | | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | NDTB[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TBSA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TBSA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 30 – TFQM** Tx FIFO/Queue Mode

| Value | Description |
|---|---|
| 0 | Tx FIFO operation. |
| 1 | Tx Queue operation. |

**Bits 29:24 – TFQS[5:0]** Transmit FIFO/Queue Size

| Value | Description |
|---|---|
| 0 | No Tx FIFO/Queue. |
| 1 – 32 | Number of Tx Buffers used for Tx FIFO/Queue. |
| >32 | Values greater than 32 are interpreted as 32. |

**Bits 21:16 – NDTB[5:0]** Number of Dedicated Transmit Buffers

| Value | Description |
|---|---|
| 0 | No Tx FIFO/Queue. |
| 1 – 32 | Number of Tx Buffers used for Tx FIFO/Queue. |
| >32 | Values greater than 32 are interpreted as 32. |

**Bits 15:0 – TBSA[15:0]** Tx Buffers Start Address
Start address of Tx Buffers section in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 39.8.36. Tx FIFO/Queue Status

**Name:** TXFQS
**Offset:** 0xC4
**Reset:** 0x00000000
**Property:** Read-only

**Note:** In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indexes indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | TFQF | TFQPI[4:0] | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | TFGI[4:0] | | | | |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | TFFL[5:0] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 21 – TFQF** Tx FIFO/Queue Full

| Value | Description |
|---|---|
| 0 | Tx FIFO/Queue not full. |
| 1 | Tx FIFO/Queue full. |

**Bits 20:16 – TFQPI[4:0]** Tx FIFO/Queue Put Index
Tx FIFO/Queue write index pointer, range 0 to 31.

**Bits 12:8 – TFGI[4:0]** Tx FIFO/Queue Get Index
Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1').

**Bits 5:0 – TFFL[5:0]** Tx FIFO Free Level
Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC.TFQM = '1').

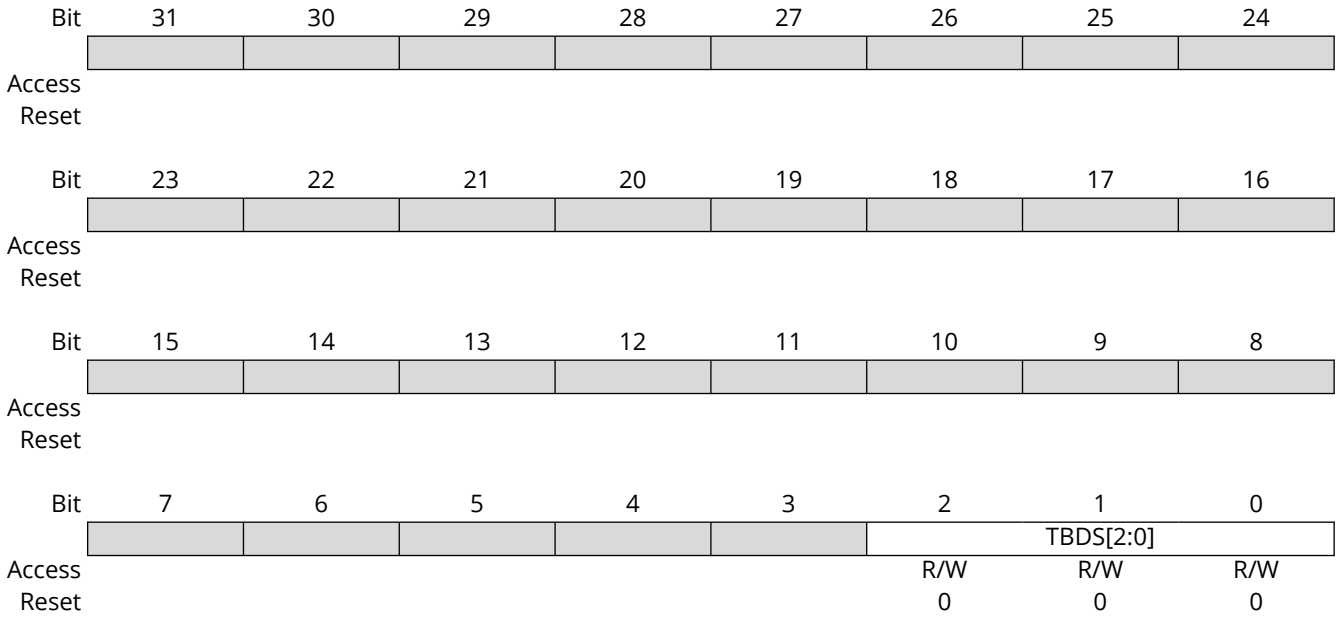

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.37. Tx Buffer Element Size Configuration

**Name:** TXESC
**Offset:** 0xC8
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes >8 bytes are intended for CAN FD operation only.




**Bits 2:0 – TBDS[2:0]** Tx Buffer Data Field Size
In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC.TBDS, the bytes not defined by the Tx Buffer are transmitted as “0xCC” (padding bytes).

| Value | Name | Description |
|---|---|---|
| 0x0 | DATA8 | 8 byte data field. |
| 0x1 | DATA12 | 12 byte data field. |
| 0x2 | DATA16 | 16 byte data field. |
| 0x3 | DATA20 | 20 byte data field. |
| 0x4 | DATA24 | 24 byte data field. |
| 0x5 | DATA32 | 32 byte data field. |
| 0x6 | DATA48 | 48 byte data field. |
| 0x7 | DATA64 | 64 byte data field. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.38. Tx Buffer Request Pending

**Name:** TXBRP
**Offset:** 0xCC
**Reset:** 0x00000000
**Property:** Read-only

**Note:** TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is canceled immediately, the corresponding TXBRP bit is reset.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TRP31 | TRP30 | TRP29 | TRP28 | TRP27 | TRP26 | TRP25 | TRP24 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TRP23 | TRP22 | TRP21 | TRP20 | TRP19 | TRP18 | TRP17 | TRP16 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TRP15 | TRP14 | TRP13 | TRP12 | TRP11 | TRP10 | TRP9 | TRP8 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TRP7 | TRP6 | TRP5 | TRP4 | TRP3 | TRP2 | TRP1 | TRP0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TRPn** Transmission Request Pending

Each Tx Buffer has its own Transmission Request Pending bit.
The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR.
TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).
A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signaled via TXBCF

- after successful transmission together with the corresponding TXBTO bit
- when the transmission has not yet been started at the point of cancellation
- when the transmission has been aborted due to lost arbitration
- when an error occurred during frame transmission

In DAR mode all transmissions are automatically canceled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions.

| Value | Description |
|---|---|
| 0 | No transmission request pending. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Transmission request pending. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 39.8.39. Tx Buffer Add Request

**Name:** TXBAR
**Offset:** 0xD0
**Reset:** 0x00000000
**Property:** -

**Note:** If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit is already set), this add request is ignored.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | AR31 | AR30 | AR29 | AR28 | AR27 | AR26 | AR25 | AR24 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | AR23 | AR22 | AR21 | AR20 | AR19 | AR18 | AR17 | AR16 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | AR15 | AR14 | AR13 | AR12 | AR11 | AR10 | AR9 | AR8 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | AR7 | AR6 | AR5 | AR4 | AR3 | AR2 | AR1 | AR0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – ARn** Add Request

Each Tx Buffer has its own Add Request bit.
Writing a '1' will set the corresponding Add Request bit; writing a '0' has no impact. This enables the Host to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.40. Tx Buffer Cancellation Request

**Name:** TXBCR
**Offset:** 0xD4
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CR31 | CR30 | CR29 | CR28 | CR27 | CR26 | CR25 | CR24 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CR23 | CR22 | CR21 | CR20 | CR19 | CR18 | CR17 | CR16 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CR15 | CR14 | CR13 | CR12 | CR11 | CR10 | CR9 | CR8 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CR7 | CR6 | CR5 | CR4 | CR3 | CR2 | CR1 | CR0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CRn** Cancellation Request

Each Tx Buffer has its own Cancellation Request bit.
Writing a '1' will set the corresponding Cancellation Request bit; writing a '0' has no impact. This enables the Host to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset.

| Value | Description |
|---|---|
| 0 | No cancellation pending. |
| 1 | Cancellation pending. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.41. Tx Buffer Transmission Occurred

**Name:** TXBTO
**Offset:** 0xD8
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TO31 | TO30 | TO29 | TO28 | TO27 | TO26 | TO25 | TO24 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TO23 | TO22 | TO21 | TO20 | TO19 | TO18 | TO17 | TO16 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TO15 | TO14 | TO13 | TO12 | TO11 | TO10 | TO9 | TO8 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TO7 | TO6 | TO5 | TO4 | TO3 | TO2 | TO1 | TO0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TOn** Transmission Occurred

Each Tx Buffer has its own Transmission Occurred bit.
The bits are set when the corresponding TXBRP bit is cleared after a successful transmission.
The bits are reset when a new transmission is requested by writing '1' to the corresponding bit of register TXBAR.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.42. Tx Buffer Cancellation Finished

**Name:** TXBCF
**Offset:** 0xDC
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CF31 | CF30 | CF29 | CF28 | CF27 | CF26 | CF25 | CF24 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CF23 | CF22 | CF21 | CF20 | CF19 | CF18 | CF17 | CF16 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CF15 | CF14 | CF13 | CF12 | CF11 | CF10 | CF9 | CF8 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CF7 | CF6 | CF5 | CF4 | CF3 | CF2 | CF1 | CF0 |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CFn** Cancellation Finished

Each Tx Buffer has its own Cancellation Finished bit.
The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately.
The bits are reset when a new transmission is requested by writing '1' to the corresponding bit of register TXBAR.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.43. Tx Buffer Transmission Interrupt Enable

**Name:** TXBTIE
**Offset:** 0xE0
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TIE31 | TIE30 | TIE29 | TIE28 | TIE27 | TIE26 | TIE25 | TIE24 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TIE23 | TIE22 | TIE21 | TIE20 | TIE19 | TIE18 | TIE17 | TIE16 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TIE15 | TIE14 | TIE13 | TIE12 | TIE11 | TIE10 | TIE9 | TIE8 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TIE7 | TIE6 | TIE5 | TIE4 | TIE3 | TIE2 | TIE1 | TIE0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – TIEn** Transmission Interrupt Enable

Each Tx Buffer has its own Transmission Interrupt Enable bit.

| Value | Description |
|---|---|
| 0 | Transmission interrupt disabled. |
| 1 | Transmission interrupt enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.44. Tx Buffer Cancellation Finished Interrupt Enable

**Name:** TXBCIE
**Offset:** 0xE4
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CFIE31 | CFIE30 | CFIE29 | CFIE28 | CFIE27 | CFIE26 | CFIE25 | CFIE24 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CFIE23 | CFIE22 | CFIE21 | CFIE20 | CFIE19 | CFIE18 | CFIE17 | CFIE16 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CFIE15 | CFIE14 | CFIE13 | CFIE12 | CFIE11 | CFIE10 | CFIE9 | CFIE8 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CFIE7 | CFIE6 | CFIE5 | CFIE4 | CFIE3 | CFIE2 | CFIE1 | CFIE0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – CFIEn** Cancellation Finished Interrupt Enable
Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.

| Value | Description |
|---|---|
| 0 | Cancellation finished interrupt disabled. |
| 1 | Cancellation finished interrupt enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.45. Tx Event FIFO Configuration

**Name:** TXEFC
**Offset:** 0xF0
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | EFWM[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | EFS[5:0] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EFSA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EFSA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:24 – EFWM[5:0]** Event FIFO Watermark

| Value | Description |
|---|---|
| 0 | Watermark interrupt disabled. |
| 1 – 32 | Level for Tx Event FIFO watermark interrupt (IR.TEFW). |
| >32 | Watermark interrupt disabled. |

**Bits 21:16 – EFS[5:0]** Event FIFO Size
The Tx Event FIFO elements are indexed from 0 to EFS - 1.

| Value | Description |
|---|---|
| 0 | Tx Event FIFO disabled |
| 1 – 32 | Number of Tx Event FIFO elements. |
| >32 | Values greater than 32 are interpreted as 32. |

**Bits 15:0 – EFSA[15:0]** Event FIFO Start Address
Start address of Tx Event FIFO in Message RAM. When the CAN module addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses, i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored. Bits 1 to 0 will always be read back as “00”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.46. Tx Event FIFO Status

**Name:** TXEFS
**Offset:** 0xF4
**Reset:** 0x00000000
**Property:** Read-only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | TEFL | EFF |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | EFPI[4:0] | | | | |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | EFGI[4:0] | | | | |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | EFFL[5:0] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 25 – TEFL** Tx Event FIFO Element Lost
This bit is a copy of interrupt flag IR.TEFL. When IR.TEFL is reset, this bit is also reset.

| Value | Description |
|---|---|
| 0 | No Tx Event FIFO element lost. |
| 1 | Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero. |

**Bit 24 – EFF** Event FIFO Full

| Value | Description |
|---|---|
| 0 | Tx Event FIFO not full. |
| 1 | Tx Event FIFO full. |

**Bits 20:16 – EFPI[4:0]** Event FIFO Put Index
Tx Event FIFO write index pointer, range 0 to 31.

**Bits 12:8 – EFGI[4:0]** Event FIFO Get Index
Tx Event FIFO read index pointer, range 0 to 31.

**Bits 5:0 – EFFL[5:0]** Event FIFO Fill Level
Number of elements stored in Tx Event FIFO, range 0 to 32.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.8.47. Tx Event FIFO Acknowledge

**Name:** TXEFA
**Offset:** 0xF8
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | EFAI[4:0] | | | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bits 4:0 – EFAI[4:0]** Event FIFO Acknowledge Index
After the Host has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS.EFGI to EFAI + 1 and update the FIFO 0 Fill Level TXEFS.EFFL.

## 39.9. Message RAM

For storage of Rx/Tx messages and for storage of the filter configuration a single- or dual-ported Message RAM has to be connected to the CAN module.

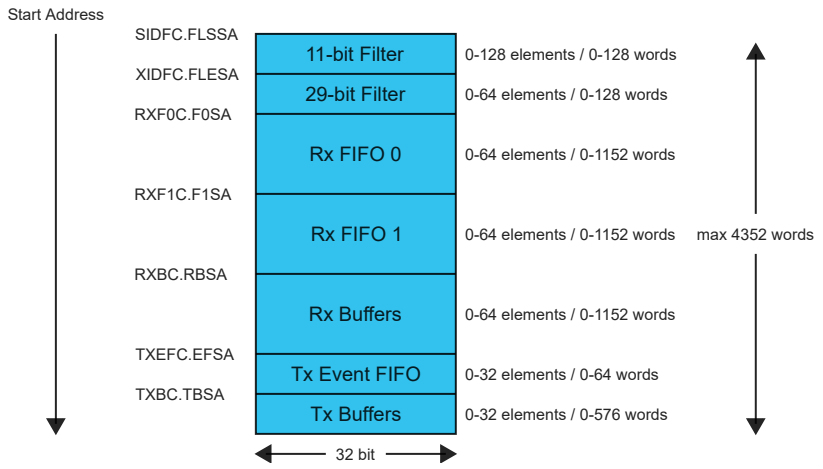
### 39.9.1. Message RAM Configuration

The Message RAM has a width of 32 bits. In case parity checking or ECC is used a respective number of bits has to be added to each word. The CAN module can be configured to allocate up to 4352 words in the Message RAM. It is not necessary to configure each of the sections listed in the figure below, nor is there any restriction with respect to the sequence of the sections.

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO 0, Rx FIFO 1, Rx Buffers, and Tx Buffers via RXESC.F0DS, RXESC.F1DS, RXESC.RBDS, and TXESC.TBDS.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 39-12.** Message RAM Configuration




When the CAN addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses (i.e., only bits 15 to 2 are evaluated and the two LSBs are ignored).

**Note:** Message RAM can only be located in the first 64 KB area of the system RAM.

**WARNING** The CAN does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.

## 39.9.2. Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in the table below. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC.

**Table 39-8.** Rx Buffer and FIFO Element

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R0 | ESI | XTD | RTR | ID[28:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| R1 | ANMF | FIDX[6:0] | | | | | | | | | FDF | BRS | DLC[3:0] | | | | RXTS[15:0] | | | | | | | | | | | | | | | |
| R2 | DB3[7:0] | | | | | | | | DB2[7:0] | | | | | | | | DB1[7:0] | | | | | | | | DB0[7:0] | | | | | | | |
| R3 | DB7[7:0] | | | | | | | | DB6[7:0] | | | | | | | | DB5[7:0] | | | | | | | | DB4[7:0] | | | | | | | |
| ... | ... | | | | | | | | ... | | | | | | | | ... | | | | | | | | ... | | | | | | | |
| Rn | DBm[7:0] | | | | | | | | DBm-1[7:0] | | | | | | | | DBm-2[7:0] | | | | | | | | DBm-3[7:0] | | | | | | | |

- R0 Bit 31 - ESI: Error State Indicator
  0 : Transmitting node is error active.
  1 : Transmitting node is error passive.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- R0 Bit 30 - XTD: Extended Identifier
  Signals to the Host whether the received frame has a standard or extended identifier.
  0 : 11-bit standard identifier.
  1 : 29-bit extended identifier.
- R0 Bit 29 - RTR: Remote Transmission Request
  Signals to the Host whether the received frame is a data frame or a remote frame.
  0 : Received frame is a data frame.
  1 : Received frame is a remote frame.
  **Note:** There are no remote frames in CAN FD format. In case a CAN FD frame was received (EDL = '1'), bit RTR reflects the state of the reserved bit r1.
- R0 Bits 28:0 - ID[28:0]: Identifier
  Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
- R1 Bit 31 - ANMF: Accepted Non-matching Frame
  Acceptance of non-matching frames may be enabled via GFC.ANFS and GFC.ANFE.
  0 : Received frame matching filter index FIDX.
  1 : Received frame did not match any Rx filter element.
- R1 Bits 30:24 - FIDX[6:0]: Filter Index
  0-127 : Index of matching Rx acceptance filter element (invalid if ANMF = '1').
  **Note:** Range is 0 to SIDFC.LSS-1 for standard and 0 to XIDFC.LSE-1 for extended.
- R1 Bits 23:22 - Reserved
- R1 Bit 21 - FDF: FD Format
  0 : Standard frame format.
  1 : CAN FD frame format (new DLC-coding and CRC).
- R1 Bit 20 - BRS: Bit Rate Search
  0 : Frame received without bit rate switching.
  1 : Frame received with bit rate switching.
- R1 Bits 19:16 - DLC[3:0]: Data Length Code
  0-8 : CAN + CAN FD: received frame has 0-8 data bytes.
  9-15 : CAN: received frame has 8 data bytes.
  9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.
- R1 Bits 15:0 - RXTS[15:0]: Rx Timestamp
  Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP.
- R2 Bits 31:24 - DB3[7:0]: Data Byte 3
- R2 Bits 23:16 - DB2[7:0]: Data Byte 2
- R2 Bits 15:8 - DB1[7:0]: Data Byte 1
- R2 Bits 7:0 - DB0[7:0]: Data Byte 0
- R3 Bits 31:24 - DB7[7:0]: Data Byte 7
- R3 Bits 23:16 - DB6[7:0]: Data Byte 6
- R3 Bits 15:8 - DB5[7:0]: Data Byte 5
- R3 Bits 7:0 - DB4[7:0]: Data Byte 4


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

...

- Rn Bits 31:24 - DBm[7:0]: Data Byte m
- Rn Bits 23:16 - DBm-1[7:0]: Data Byte m-1
- Rn Bits 15:8 - DBm-2[7:0]: Data Byte m-2
- Rn Bits 7:0 - DBm-3[7:0]: Data Byte m-3

**WARNING** Depending on the configuration of RXESC, between two and sixteen 32-bit words (Rn = 3 ... 17) are used for storage of a CAN message's data field.

### 39.9.3. Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

**Table 39-9.** Tx Buffer Element

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T0 | ESI | XTD | RTR | ID[28:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| T1 | MM[7:0] | | | | | | | | EFC | | FDF | BRS | DLC[3:0] | | | | | | | | | | | | | | | | | | | |
| T2 | DB3[7:0] | | | | | | | | DB2[7:0] | | | | | | | | DB1[7:0] | | | | | | | | DB0[7:0] | | | | | | | |
| T3 | DB7[7:0] | | | | | | | | DB6[7:0] | | | | | | | | DB5[7:0] | | | | | | | | DB4[7:0] | | | | | | | |
| ... | ... | | | | | | | | ... | | | | | | | | ... | | | | | | | | ... | | | | | | | |
| Tn | DBm[7:0] | | | | | | | | DBm-1[7:0] | | | | | | | | DBm-2[7:0] | | | | | | | | DBm-3[7:0] | | | | | | | |

- T0 Bit 31 - ESI: Error State Indicator

  0 : ESI bit in CAN FD format depends only on error passive flag.

  1 : ESI bit in CAN FD format transmitted recessive.

  **Note:** The ESI bit of the transmit buffer is OR'ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.
- T0 Bit 30 - XTD: Extended Identifier

  0 : 11-bit standard identifier.

  1 : 29-bit extended identifier.
- T0 Bit 29 - RTR: Remote Transmission Request

  0 : Transmit data frame.

  1 : Transmit remote frame.

  **Note:** When RTR = '1', the CAN transmits a remote frame according to ISO 11898-1, even if CCCR.CME enables the transmission in CAN FD format.
- T0 Bits 28:0 - ID[28:0]: Identifier


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].

- T1 Bits 31:24 - MM[7:0]: Message Marker
  Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.
- T1 Bit 23 - EFC: Event FIFO Control
  0 : Don't store Tx events.
  1 : Store Tx events.
- T1 Bit 22 - Reserved
- TR1 Bit 21 - FDF: FD Format
  0 : Frame transmitted in Classic CAN format.
  1 : Frame transmitted in CAN FD format.
- T1 Bit 20 - BRS: Bit Rate Search
  0 : CAN FD frames transmitted without bit rate switching.
  1 : CAN FD frames transmitted with bit rate switching.
  **Note:** Bits ESI, FDF, and BRS are only evaluated when CAN FD operation is enabled CCCR.FDOE = '1'. Bit BRS is only evaluated when in addition CCCR.BRSE = '1'.
- T1 Bits 19:16 - DLC[3:0]: Data Length Code
  0-8 : CAN + CAN FD: received frame has 0-8 data bytes.
  9-15 : CAN: received frame has 8 data bytes.
  9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.
- T1 Bits 15:0 - Reserved
- T2 Bits 31:24 - DB3[7:0]: Data Byte 3
- T2 Bits 23:16 - DB2[7:0]: Data Byte 2
- T2 Bits 15:8 - DB1[7:0]: Data Byte 1
- T2 Bits 7:0 - DB0[7:0]: Data Byte 0
- T3 Bits 31:24 - DB7[7:0]: Data Byte 7
- T3 Bits 23:16 - DB6[7:0]: Data Byte 6
- T3 Bits 15:8 - DB5[7:0]: Data Byte 5
- T3 Bits 7:0 - DB4[7:0]: Data Byte 4
  ...
- Tn Bits 31:24 - DBm[7:0]: Data Byte m
- Tn Bits 23:16 - DBm-1[7:0]: Data Byte m-1
- Tn Bits 15:8 - DBm-2[7:0]: Data Byte m-2
- Tn Bits 7:0 - DBm-3[7:0]: Data Byte m-3

**Note:** Depending on the configuration of TXESC, between two and sixteen 32-bit words (Tn = 3 ... 17) are used for storage of a CAN message's data field.

### 39.9.4. Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 39-10.** Tx Event FIFO Element

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E0 | ESI | XTD | RTR | ID[28:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| E1 | MM[7:0] | | | | | | | | ET[1:0] | | FDF | BRS | DLC[3:0] | | | | TXTS[15:0] | | | | | | | | | | | | | | | |

- E0 Bit 31 - ESI: Error State Indicator
  0 : Transmitting node is error active.
  1 : Transmitting node is error passive.
- E0 Bit 30 - XTD: Extended Identifier
  0 : 11-bit standard identifier.
  1 : 29-bit extended identifier.
- E0 Bit 29 - RTR: Remote Transmission Request
  0 : Received frame is a data frame.
  1 : Received frame is a remote frame.
- E0 Bits 28:0 - ID[28:0]: Identifier
  Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].
- E1 Bits 31:24 - MM[7:0]: Message Marker
  Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.
- E1 Bits 23:22 - ET[1:0]: Event Type
  This field defines the event type.

**Table 39-11.** Event Type

| Value | Name | Description |
|---|---|---|
| 0x0 or 0x3 | RES | Reserved |
| 0x1 | TXE | Tx event |
| 0x2 | TXC | Transmission in spite of cancellation (always set for transmission in DAR mode) |

- E1 Bit 21 - FDF: FD Format
  0 : Standard frame format.
  1 : CAN FD frame format (new DLC-coding and CRC).
- E1 Bit 20 - BRS: Bit Rate Search
  0 : Frame received without bit rate switching.
  1 : Frame received with bit rate switching.
- E1 Bits 19:16 - DLC[3:0]: Data Length Code
  0-8 : CAN + CAN FD: received frame has 0-8 data bytes.
  9-15 : CAN: received frame has 8 data bytes.
  9-15 : CAN FD: received frame has 12/16/20/24/32/48/64 data bytes.
- E1 Bits 15:0 - TXTS[15:0]: Tx Timestamp
  Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.9.5. Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address SIDFC.FLSSA plus the index of the filter element (0 ... 127).

**Table 39-12.** Standard Message ID Filter Element

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S0 | SFT[1:0] | | SFEC[2:0] | | | SFID1[10:0] | | | | | | | | | | | | | | | | SFID2[10:0] | | | | | | | | | | |

- Bits 31:30 - SFT[1:0]: Standard Filter Type
  This field defines the standard filter type.

**Table 39-13.** Standard Filter Type

| Value | Name | Description |
|---|---|---|
| 0x0 | RANGE | Range filter from SFID1 to SFID2 (SFID2 >= SFID1) |
| 0x1 | DUAL | Dual ID filter for SFID1 or SFID2 |
| 0x2 | CLASSIC | Classic filter: SFID1 = filter, SFID2 = mask |
| 0x3 | RES | Reserved |

- Bits 29:27 - SFEC[2:0]: Standard Filter Element Configuration
  All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101”, or “110” a match sets interrupt flag IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.

**Table 39-14.** Standard Filter Element Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Disable filter element |
| 0x1 | STF0M | Store in Rx FIFO 0 if filter matches |
| 0x2 | STF1M | Store in Rx FIFO 1 if filter matches |
| 0x3 | REJECT | Reject ID if filter matches |
| 0x4 | PRIORITY | Set priority if filter matches. |
| 0x5 | PRIF0M | Set priority and store in FIFO 0 if filter matches. |
| 0x6 | PRIF1M | Set priority and store in FIFO 1 if filter matches. |
| 0x7 | STRXBUF | Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored. |

- Bits 26:16 - SFID1[10:0]: Standard Filter ID 1
  First ID of standard ID filter element.
  When filtering for Rx Buffers or for debug messages this field defines the ID of a standard mesage to be stored. The received identifiers must match exactly, no masking mechanism is used.
- Bits 15:11 - Reserved
- Bits 10:0 - SFID2[10:0]: Standard Filter ID 2
  This bit field has a different meaning depending on the configuration of SFEC.
  a. SFEC = “001” ... “110”: Second ID of standard ID filter element.
  b. SFEC = “111”: Filter for Rx Buffers or for debug messages.
  SFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.
  00 = Store message into an Rx Buffer


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

01 = Debug Message A

10 = Debug Message B

11 = Debug Message C

SFID2[8:6] is Reserved.

SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message.

### 39.9.6. Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address XIDFC.FLESA plus two times the index of the filter element (0...63).

**Table 39-15.** Extended Message ID Filter Element

| | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F0 | EFEC[2:0] | | | EFID1[28:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| F1 | EFT[1:0] | | | EFID2[28:0] | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

- F0 Bits 31:29 - EFEC[2:0]: Extended Filter Element Configuration

  All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110” a match sets interrupt flag IR.HPM and, if enabled, an interrupt is generated. In this case the HPMS register is updated with the status of the priority match.

**Table 39-16.** Extended Filter Element Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Disable filter element. |
| 0x1 | STF0M | Store in Rx FIFO 0 if filter matches. |
| 0x2 | STF1M | Store in Rx FIFO 1 if filter matches. |
| 0x3 | REJECT | Reject ID if filter matches. |
| 0x4 | PRIORITY | Set priority if filter matches. |
| 0x5 | PRIF0M | Set priority and store in FIFO 0 if filter matches. |
| 0x6 | PRIF1M | Set priority and store in FIFO 1 if filter matches. |
| 0x7 | STRXBUF | Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored. |

- F0 Bits 28:0 - EFID1[28:0]: Extended Filter ID 1

  First ID of extended ID filter element.

  When filtering for Rx Buffers or for debug messages this field defines the ID of a extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism is used.

- F1 Bits 31:30 - EFT[1:0]: Extended Filter Type

  This field defines the extended filter type.

**Table 39-17.** Extended Filter Type

| Value | Name | Description |
|---|---|---|
| 0x0 | RANGEM | Range filter from EFID1 to EFID2 (EFID2 >= EFID1). |
| 0x1 | DUAL | Dual ID filter for EFID1 or EFID2. |
| 0x2 | CLASSIC | Classic filter: EFID1 = filter, EFID2 = mask. |
| 0x3 | RANGE | Range filter from EFID1 to EFID2 (EFID2 >= EFID1), XIDAM mask not applied. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- F1 Bits 28:0 - EFID2[28:0]: Extended Filter ID 2

  This bit field has a different meaning depending on the configuration of EFEC.
  1) EFEC = “001” ... “110” Second ID of standard ID filter element.
  2) EFEC = “111” Filter for Rx Buffers or for debug messages.

  EFID2 [10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

  00 = Store message into an Rx Buffer

  01 = Debug Message A

  10 = Debug Message B

  11 = Debug Message C

  EFID2 [8:6] is used to control the filter event pins at the Extension Interface. A ‘1’ at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one CLK_CANx_AHB period in case the filter matches.

  EFID2 [5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 40. SD/MMC Host Controller (SDHC)

## 40.1. Overview

The SD/MMC Host Controller (SDHC) supports the embedded MultiMedia Card (e.MMC) Specification, the SD Memory Card Specification, and the SDIO Specification. It is compliant with the SD Host Controller Standard specifications. Refer to Reference Documents for details.

The SDHC includes the register set defined in the “SD Host Controller Simplified Specification V3.00” and additional registers to manage e.MMC devices and enhanced features.

The SDHC is clocked by up to three clocks (bus clock, SDHC core clock, and a slow clock for certain functions). Both the MCLK and GCLK must be configured before the SDHC can be used.

The SAM D5x/E5x provides two instances of the SDHC, SDHC0 and SDHC1.

**Related Links**
Block Diagram

### 40.1.1. Reference Documents

| Name | Link |
|---|---|
| SD Host Controller Simplified Specification V3.00 | www.sdcard.org |
| SDIO Simplified Specification V3.00 | |
| Physical Layer Simplified Specification V3.01 | |
| Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51 | www.jedec.org |

## 40.2. Features

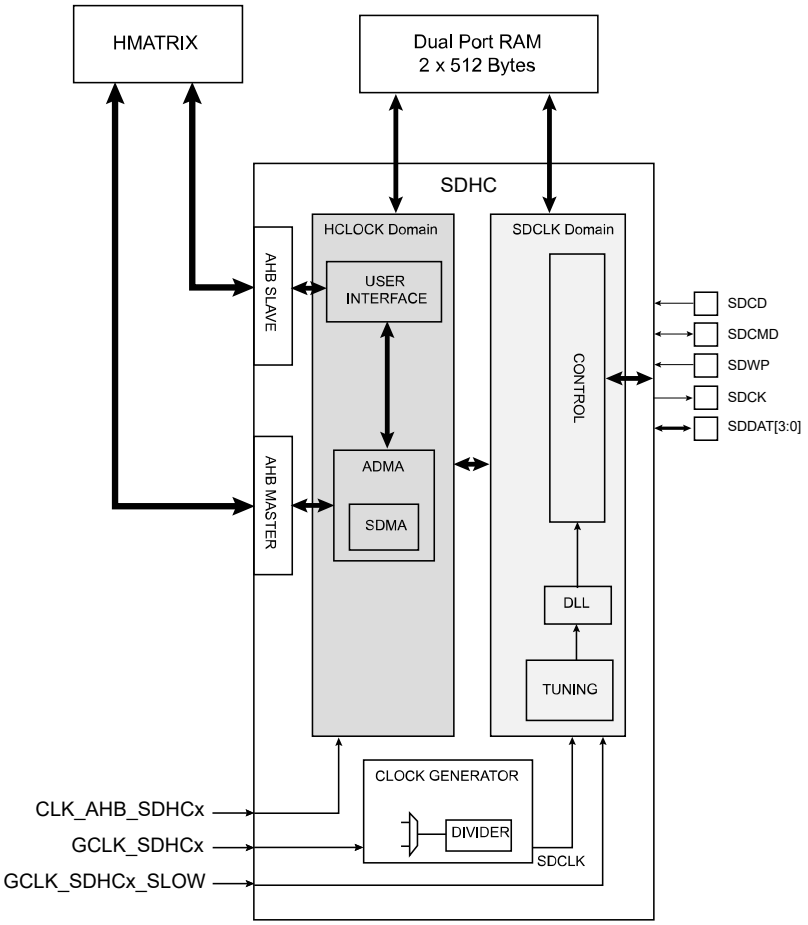
- Compatibility:
  - SD Host Controller Standard Specification
  - MultiMedia Card Specification
  - SD Memory Card Specification
  - SDIO Specification Version

  Refer to Reference Documents for details.
- Support for 1-bit/ 4-bit SD/SDIO Devices
- Support for 1-bit/4-bit e.MMC Devices
- Support for SD/SDIO Default Speed (Maximum SDCLK Frequency = 25 MHz)
- Support for SD/SDIO High Speed (Maximum SDCLK Frequency = 50 MHz)
- Support for e.MMC Default Speed (Maximum SDCLK Frequency = 26 MHz)
- e.MMC Boot Operation Mode Support
- Support for Block Size from 1 to 512 bytes
- Support for Stream, Block and Multi-block Data Read and Write – Advanced DMA and SDMA Capability
- Internal 1024-byte Dual Port RAM
- Support for both synchronous and asynchronous abort
- Supports for SDIO Card Interrupt

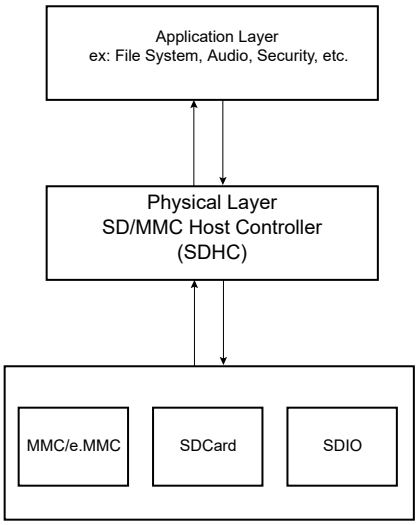

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.3. Block Diagrams

### 40.3.1. Block Diagram




### 40.3.2. Application Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.4. Signal Description

| Signal Name | Type | Description |
|---|---|---|
| SDCD | Input | SD Card / SDIO /e.MMC Card Detect |
| SDCMD | I/O | SD Card / SDIO /e.MMC Command/ Response Line |
| SDWP | Input | SD Card Connector Write Protect Signal |
| SDCK | Output | SD Card / SDIO /e.MMC Clock Signal |
| SDDAT[3:0] | I/O | SD Card / SDIO /e.MMC data lines |

## 40.5. Product Dependencies

### 40.5.1. I/O Lines

In order to use the I/O lines, the I/O pins must be configured using the IO Pin Controller (PORT).

### 40.5.2. Power Management

This peripheral can continue to operate in any sleep mode where its source clock is running. Refer to PM – Power Manager for details on the different sleep modes.

### 40.5.3. Clocks

The peripheral is using two generic clocks and one bus clock.

The clock for the SDHC bus interface (CLK_AHB_SDHC) is enabled and disabled by the Main Clock Controller. The default state of CLK_AHB_SDHC can be found in the Peripheral Clock Masking section.

The two generic clocks are:

- The core clock GCLK_SDHCx is required to clock the SDHC core.
- The slow clock GCLK_SDHCx_SLOW is only required for certain functions. When this clock is required, GCLK_SDHCx must be enabled.

These clocks must be configured and enabled in the Generic Clock Controller (GCLK) before using the SDHC. The generic clocks are asynchronous to the user interface clock (CLK_SDHCx_AHB).

**Related Links**

MCLK – Main Clock

### 40.5.4. DMA

Not applicable.

### 40.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first.

### 40.5.6. Events

Not applicable.

## 40.6. Functional Description

### 40.6.1. SD/SDIO Operating Mode

This peripheral is fully compliant with the "SD Host Controller Simplified Specification V3.00" for SD/SDIO devices. Refer to this specification for configuration.

Refer to "Physical Layer Simplified Specification V3.01" and "SDIO Simplified Specification V3.00" for SD/SDIO management.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Reference Documents

### 40.6.2. e.MMC Operating Mode

This peripheral supports e.MMC devices management. As the “SD Host Controller Simplified Specification V3.00” does not apply to e.MMC devices, some registers have been added to those described in this specification in order to manage e.MMC devices. Most of the registers described in the “SD Host Controller Simplified Specification V3.00” must be used for e.MMC management, but e.MMC-specific features are managed using MC1R and MC2R.

**Related Links**
Reference Documents


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 40.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | SSAR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x04 | BSR | 7:0 | BLKSIZE[7:0] | | | | | | | |
| | | 15:8 | | BOUNDARY[2:0] | | | | | BLKSIZE[9:8] | |
| 0x06 | BCR | 7:0 | BLKCNT[7:0] | | | | | | | |
| | | 15:8 | BLKCNT[15:8] | | | | | | | |
| 0x08 | ARG1R | 7:0 | ARG1[7:0] | | | | | | | |
| | | 15:8 | ARG1[15:8] | | | | | | | |
| | | 23:16 | ARG1[23:16] | | | | | | | |
| | | 31:24 | ARG1[31:24] | | | | | | | |
| 0x0C | TMR | 7:0 | | | MSBSEL | DTDSEL | ACMDEN[1:0] | | BCEN | DMAEN |
| | | 15:8 | | | | | | | | |
| 0x0E | CR | 7:0 | CMDTYP[1:0] | | DPSEL | CMDICEN | CMDCCEN | | RESPTYP[1:0] | |
| | | 15:8 | | | CMDIDX[5:0] | | | | | |
| 0x10 | RR0 | 7:0 | CMDRESP[7:0] | | | | | | | |
| | | 15:8 | CMDRESP[15:8] | | | | | | | |
| | | 23:16 | CMDRESP[23:16] | | | | | | | |
| | | 31:24 | CMDRESP[31:24] | | | | | | | |
| 0x14 | RR1 | 7:0 | CMDRESP[7:0] | | | | | | | |
| | | 15:8 | CMDRESP[15:8] | | | | | | | |
| | | 23:16 | CMDRESP[23:16] | | | | | | | |
| | | 31:24 | CMDRESP[31:24] | | | | | | | |
| 0x18 | RR2 | 7:0 | CMDRESP[7:0] | | | | | | | |
| | | 15:8 | CMDRESP[15:8] | | | | | | | |
| | | 23:16 | CMDRESP[23:16] | | | | | | | |
| | | 31:24 | CMDRESP[31:24] | | | | | | | |
| 0x1C | RR3 | 7:0 | CMDRESP[7:0] | | | | | | | |
| | | 15:8 | CMDRESP[15:8] | | | | | | | |
| | | 23:16 | CMDRESP[23:16] | | | | | | | |
| | | 31:24 | CMDRESP[31:24] | | | | | | | |
| 0x20 | BDPR | 7:0 | BUFDATA[7:0] | | | | | | | |
| | | 15:8 | BUFDATA[15:8] | | | | | | | |
| | | 23:16 | BUFDATA[23:16] | | | | | | | |
| | | 31:24 | BUFDATA[31:24] | | | | | | | |
| 0x24 | PSR | 7:0 | | | | | RTREQ | DLACT | CMDINHD | CMDINHC |
| | | 15:8 | | | | | BUFRDEN | BUFWREN | RTACT | WTACT |
| | | 23:16 | DATLL[3:0] | | | | WRPPL | CARDDPL | CARDSS | CARDINS |
| | | 31:24 | | | | | | | | CMDLL |
| 0x28 | HC1R | 7:0 | CARDDSEL | CARDDTL | | DMASEL[1:0] | | HSEN | DW | LEDCTRL |
| 0x29 | PCR | 7:0 | | | | | | SDBVSEL[2:0] | | SDBPWR |
| 0x2A | BGCR | 7:0 | | | | | INTBG | RWCTRL | CONTR | STPBGR |
| 0x2B | WCR | 7:0 | | | | | | WKENCREM | WKENCINS | WKENCINT |
| 0x2C | CCR | 7:0 | USDCLKFSEL[1:0] | | CLKGSEL | | | SDCLKEN | INTCLKS | INTCLKEN |
| | | 15:8 | SDCLKFSEL[7:0] | | | | | | | |
| 0x2E | TCR | 7:0 | | | | | DTCVAL[3:0] | | | |
| 0x2F | SRR | 7:0 | | | | | | SWRSTDAT | SWRSTCMD | SWRSTALL |
| 0x30 | NISTR | 7:0 | CREM | CINS | BRDRDY | BWRRDY | DMAINT | BLKGE | TRFC | CMDC |
| | | 15:8 | ERRINT | BOOTAR | | | | | | CINT |
| 0x32 | EISTR | 7:0 | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| | | 15:8 | | | | BOOTAE | | | ADMA | ACMD |
| 0x34 | NISTER | 7:0 | CREM | CINS | BRDRDY | BWRRDY | DMAINT | BLKGE | TRFC | CMDC |
| | | 15:8 | | BOOTAR | | | | | | CINT |
| 0x36 | EISTER | 7:0 | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| | | 15:8 | | | | BOOTAE | | | ADMA | ACMD |
| 0x38 | NISIER | 7:0 | CREM | CINS | BRDRDY | BWRRDY | DMAINT | BLKGE | TRFC | CMDC |
| | | 15:8 | | BOOTAR | | | | | | CINT |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x3A | EISIER | 7:0 | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| | | 15:8 | | | | BOOTAE | | | ADMA | ACMD |
| 0x3C | ACESR | 7:0 | CMDNI | | | ACMDIDX | ACMDEND | ACMDCRC | ACMDTEO | ACMD12NE |
| | | 15:8 | | | | | | | | |
| 0x3E | HC2R - EMMC | 7:0 | SCLKSEL | EXTUN | DRVSEL[1:0] | | HS200EN[3:0] | | | |
| | | 15:8 | PVALEN | | | | | | | |
| 0x3E | HC2R - DEFAULT | 7:0 | SCLKSEL | EXTUN | DRVSEL[1:0] | | VS18EN | UHSMS[2:0] | | |
| | | 15:8 | PVALEN | ASINTEN | | | | | | |
| 0x40 | CA0R | 7:0 | TEOCLKU | | TEOCLKF[5:0] | | | | | |
| | | 15:8 | BASECLKF[7:0] | | | | | | | |
| | | 23:16 | SRSUP | SDMASUP | HSSUP | | ADMA2SUP | ED8SUP | MAXBLKL[1:0] | |
| | | 31:24 | SLTYPE[1:0] | | ASINTSUP | SB64SUP | | V18VSUP | V30VSUP | V33VSUP |
| 0x44 | CA1R | 7:0 | | DRVDSUP | DRVCSUP | DRVASUP | | DDR50SUP | SDR104SUP | SDR50SUP |
| | | 15:8 | | | TSDR50 | | TCNTRT[3:0] | | | |
| | | 23:16 | CLKMULT[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x48 | MCCAR | 7:0 | MAXCUR33V[7:0] | | | | | | | |
| | | 15:8 | MAXCUR30V[7:0] | | | | | | | |
| | | 23:16 | MAXCUR18V[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C ... 0x4F | Reserved | | | | | | | | | |
| 0x50 | FERACES | 7:0 | CMDNI | | | ACMDIDX | ACMDEND | ACMDCRC | ACMDTEO | ACMD12NE |
| | | 15:8 | | | | | | | | |
| 0x52 | FEREIS | 7:0 | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| | | 15:8 | | | | BOOTAE | | | ADMA | ACMD |
| 0x54 | AESR | 7:0 | | | | | | LMIS | ERRST[1:0] | |
| 0x55 ... 0x57 | Reserved | | | | | | | | | |
| 0x58 | ASAR | 7:0 | ADMASA[7:0] | | | | | | | |
| | | 15:8 | ADMASA[15:8] | | | | | | | |
| | | 23:16 | ADMASA[23:16] | | | | | | | |
| | | 31:24 | ADMASA[31:24] | | | | | | | |
| 0x5C ... 0x5F | Reserved | | | | | | | | | |
| 0x60 | PVR0 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x62 | PVR1 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x64 | PVR2 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x66 | PVR3 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x68 | PVR4 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x6A | PVR5 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x6C | PVR6 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x6E | PVR7 | 7:0 | SDCLKFSEL[7:0] | | | | | | | |
| | | 15:8 | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| 0x70 ... 0xFB | Reserved | | | | | | | | | |
| 0xFC | SISR | 7:0 | | | | | | | INTSSL[1:0] | |
| | | 15:8 | | | | | | | | |
| 0xFE | HCVR | 7:0 | SVER[7:0] | | | | | | | |
| | | 15:8 | VVER[7:0] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0100 ... 0x01FF | Reserved | | | | | | | | | |
| 0x0200 | APSR | 7:0 | | | | | HDATLL[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0204 | MC1R | 7:0 | FCD | RSTN | BOOTA | OPD | DDR | | CMDTYP[1:0] | |
| 0x0205 | MC2R | 7:0 | | | | | | | ABOOT | SRESP |
| 0x0206 ... 0x0207 | Reserved | | | | | | | | | |
| 0x0208 | ACR | 7:0 | | | | | | | BMAX[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x020C | CC2R | 7:0 | | | | | | | | FSDCLKD |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0210 ... 0x022F | Reserved | | | | | | | | | |
| 0x0230 | CACR | 7:0 | | | | | | | | CAPWREN |
| | | 15:8 | KEY[7:0] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0234 | DBGR | 7:0 | | | | | | | | NIDBG |
| | | 15:8 | | | | | | | | |

## 40.8. Register Description


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.1. SDMA System Address / Argument 2 Register

**Name:** SSAR
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** -

This register contains the physical system memory address used for SDMA transfers or the second argument for Auto CMD23.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADDR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADDR[31:0]** SDMA System Address/Argument 2

The function of this bit field is depending on the operation mode:
For a SDMA transfer, this field is the system memory address. When the peripheral stops an SDMA transfer, this field points to the system address of the next contiguous data position. This field can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. An interrupt can be generated to instruct the software to update this field. Writing the next system address of the next data position restarts the SDMA transfer.
When executing Auto CMD23, this field is used with Auto CMD23 to set a 32-bit block count value to the CMD23 argument. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by BCR. In this case, 65535 blocks is the maximum value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.2. Block Size Register

**Name:** BSR
**Offset:** 0x04
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | BOUNDARY[2:0] | | | | | BLKSIZE[9:8] | |
| Access | | - | - | - | | | R/W | R/W |
| Reset | | 0 | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BLKSIZE[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 14:12 – BOUNDARY[2:0]** SDMA Buffer Boundary

This field specifies the size of the contiguous buffer in the system memory. The SDMA transfer waits at every boundary specified by this field and the peripheral generates the DMA Interrupt to instruct the software to update SSAR. If this field is set to 0 (buffer size = 4 Kbytes), the lowest 12 bits of SSAR.ADDRESS point to data in the contiguous buffer, and the upper 20 bits point to the location of the buffer in the system memory. This function is active when the DMA Enable bit in the Transfer Mode Register (TMR.DMAEN) is '1'.

| Value | Name | Description |
|---|---|---|
| 0 | 4K | 4-Kbyte boundary |
| 1 | 8K | 8-Kbyte boundary |
| 2 | 16K | 16-Kbyte boundary |
| 3 | 32K | 32-Kbyte boundary |
| 4 | 64K | 64-Kbyte boundary |
| 5 | 128K | 128-Kbyte boundary |
| 6 | 256k | 256-Kbyte boundary |
| 7 | 512K | 512-Kbyte boundary |

**Bits 9:0 – BLKSIZE[9:0]** Transfer Block Size

This field specifies the block size of data transfers for CMD17, CMD18, CMD24, CMD25 and CMD53. Values ranging from 1 to 512 can be set. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.3. Block Count Register

**Name:** BCR
**Offset:** 0x06
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BLKCNT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BLKCNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – BLKCNT[15:0]** Block Count for Current Transfer

This field is used only if TMR.BCEN (Block Count Enable) is set to 1 and is valid only for multiple block transfers. BLKCNT is the number of blocks to be transferred and it must be set to a value between 1 and the maximum block count. The peripheral decrements the block count after each block transfer and stops when the count reaches 0. When this field is set to 0, no data block is transferred.

This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored.

When a suspend command is completed, the number of blocks yet to be transferred can be determined by reading this register. Before issuing a resume command, the previously saved block count is restored.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.4. Argument 1 Register

**Name:** ARG1R
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** Read/Write

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ARG1[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ARG1[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ARG1[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ARG1[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ARG1[31:0]** Argument 1
This register contains the SD command argument which is specified as the bit 39-8 of Command-Format in the “Physical Layer Simplified Specification V3.01” or “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51”.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.5. Transfer Mode Register

**Name:** TMR
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** -

This register is used to control data transfers. The user shall set this register before issuing a command which transfers data (refer to bit DPSEL in CR), or before issuing a Resume command. The user must save the value of this register when the data transfer is suspended (as a result of a Suspend command) and restore it before issuing a Resume command. To prevent data loss, this register cannot be written while data transactions are in progress. Writes to this register are ignored when bit PSR.CMDINHD is '1'.

**Table 40-1.** Determining the Transfer Type

| MSBSEL | BCEN | BCR.BLKCNT | Function |
|---|---|---|---|
| 0 | Don't care | Don't care | Single Transfer |
| 1 | 0 | Don't care | Infinite Transfer |
| 1 | 1 | Not Zero | Multiple Transfer |
| 1 | 1 | Zero | Stop Multiple Transfer |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MSBSEL | DTDSEL | ACMDEN[1:0] | | BCEN | DMAEN |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 5 – MSBSEL** Multi/Single Block Selection
Write this bit to '1' when issuing multiple-block transfer commands using DAT line(s). For any other commands, write this bit to 0. If this bit is 0, it is not necessary to write BCR to '1' (refer to Table 1-4).

**Bit 4 – DTDSEL** Data Transfer Direction Selection
This bit defines the direction of the DAT lines data transfers. Write this bit to '1' to transfer data from the device (SD Card/SDIO/e.MMC) to the peripheral. Write this bit to '0' for all other commands.

| Value | Name | Description |
|---|---|---|
| 0 | WRITE | Writes data from the peripheral to the device. |
| 1 | READ | Reads data from the device to the peripheral. |

**Bits 3:2 – ACMDEN[1:0]** Auto Command Enable
Two methods can be used to stop Multiple-block read and write operation:

1. Auto CMD12: when the ACMDEN field is set to 1, the peripheral issues CMD12 automatically when the last block transfer is completed. An Auto CMD12 error is indicated to ACESR. Auto CMD12 is not enabled if the command does not require CMD12.
2. Auto CMD23: when the ACMDEN field is set to 2, the peripheral issues a CMD23 automatically before issuing a command specified in CR.

The following conditions are required to use Auto CMD23:

- A memory card that supports CMD23 (SCR[33] = 1)
- If DMA is used, it must be ADMA (SDMA not supported).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Only CMD18 or CMD25 is issued.

**Note:** The peripheral does not check the command index.

Auto CMD23 can be used with or without ADMA. By writing CR, the peripheral issues a CMD23 first and then issues a command specified by the CR.CMDIDX field. If CMD23 response errors are detected, the second command is not issued. A CMD23 error is indicated in ACESR. The CMD23 argument (32-bit block count value) is defined in SSAR.
This field determines the use of auto command functions.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLED | Auto Command Disabled |
| 1 | CMD12 | Auto CMD12 Enabled |
| 2 | CMD23 | Auto CMD23 Enabled |
| 3 | Reserved | Reserved |

**Bit 1 – BCEN** Block Count Enable

This bit is used to enable BCR, which is only relevant for multiple block transfers. When this bit is 0, BCR is disabled, which is useful when executing an infinite transfer (refer to Table 1-4). If an ADMA2 transfer is more than 65535 blocks, this bit is set to 0 and the data transfer length is designated by the Descriptor Table.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Block count is disabled |
| 1 | ENABLE | Block count is enabled |

**Bit 0 – DMAEN** DMA Enable

This bit enables the DMA functionality described in section “Supporting DMA” in “SD Host Controller Simplified Specification V3.00” . DMA can be enabled only if it is supported as indicated by the bit CA0R.ADMA2SUP. One of the DMA modes can be selected using the field HC1R.DMASEL. If DMA is not supported, this bit is meaningless and then always reads 0. When this bit is set to 1, a DMA operation begins when the user writes to the upper byte of CR.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | DMA functionality is disabled |
| 1 | ENABLE | DMA functionality is enabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.6. Command Register

**Name:** CR
**Offset:** 0x0E
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | CMDIDX[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMDTYP[1:0] | | DPSEL | CMDICEN | CMDCCEN | | RESPTYP[1:0] | |
| Access | R/W | R/W | R/W | - | R/W | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |

**Bits 13:8 – CMDIDX[5:0]** Command Index
This bit shall be set to the command number (CMD0-63, ACMD0-63) that is specified in bits 45-40 of the Command-Format in the "Physical Layer Simplified Specification V3.01", "SDIO Simplified Specification V3.00", and "Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51".

**Bits 7:6 – CMDTYP[1:0]** Command Type

| Value | Name | Description |
|---|---|---|
| 0 | NORMAL | Other commands |
| 1 | SUSPEND | CMD52 to write "Bus Suspend" in the Card Common Control Registers (CCCR) (for SDIO only) |
| 2 | RESUME | CMD52 to write "Function Select" in the Card Common Control Registers (CCCR) (for SDIO only) |
| 3 | ABORT | CMD12, CMD52 to write "I/O Abort" in the Card Common Control Registers (CCCR) (for SDIO only) |

**Bit 5 – DPSEL** Data Present Select
This bit is set to 1 to indicate that data is present and shall be transferred using the DAT lines. It is set to 0 for the following:

1. Commands using only CMD line (Ex. CMD52)
2. Commands with no data transfer but using Busy signal on DAT[0] line (Ex. CMD38)
3. Resume command

| Value | Description |
|---|---|
| 0 | No data present |
| 1 | Data present |

**Bit 4 – CMDICEN** Command Index Check Enable
If this bit is set to 1, the peripheral checks the Index field in the response to see if it has the same value as the command index. If it has not, it is reported as a Command Index Error (CMDIDX) in EISTR. If this bit is set to 0, the Index field of the response is not checked.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | The Command Index Check is disabled. |
| 1 | ENABLE | The Command Index Check is enabled. |

**Bit 3 – CMDCCEN** Command CRC Check Enable
If this bit is set to 1, the peripheral checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error (CMDCRC) in EISTR. If this bit is set to 0, the CRC field is not checked. The position of the CRC field is determined according to the length of the response.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | The Command CRC Check is disabled. |
| 1 | ENABLE | The Command CRC Check is enabled. |

**Bits 1:0 – RESPTYP[1:0]** Response Type
This field is set according to the response type expected for the command index (CMDIDX).

| Value | Name | Description |
|---|---|---|
| 0 | NORESP | No Response |
| 1 | RL136 | Response Length 136 |
| 2 | RL48 | Response Length 48 |
| 3 | RL48BUSY | Response Length 48 with Busy |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 40.8.7. Response Register n

**Name:** RR
**Offset:** 0x10 + n*0x04 [n=0..3]
**Reset:** 0x000000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CMDRESP[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CMDRESP[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CMDRESP[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMDRESP[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CMDRESP[31:0]** Command Response

The table below describes the mapping of command responses from the SD/SDIO/e.MMC bus to these registers for each responses type. In this table, R[] refers to a bit range of the response data as transmitted on the SD/SDIO/e.MMC bus.

| Type of response | Meaning of response | Response field | Response register |
|---|---|---|---|
| R1, R1b (normal response) | Card Status | R[39:8] | RR0[31:0] |
| R1b (Auto CMD12 response) | Card Status for Auto CMD12 | R[39:8] | RR3[31:0] |
| R1 (Auto CMD23 response) | Card Status for Auto CMD23 | R[39:8] | RR3[31:0] |
| R2 (CID, CSD register) | CID or CSD register | R[127:8] | RR0[31:0]<br>RR1[31:0]<br>RR2[31:0]<br>RR3[23:0] |
| R3 (OCR register) | OCR register for memory | R[39:8] | RR0[31:0] |
| R4 (OCR register) | OCR register for I/O | R[39:8] | RR0[31:0] |
| R5, R5b | SDIO response | R[39:8] | RR0[31:0] |
| R6 (Published RCA response) | New published RCA[31:16] and Card status bits | R[39:8] | RR0[31:0] |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.8. Buffer Data Port Register

**Name:** BDPR
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BUFDATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BUFDATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BUFDATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BUFDATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – BUFDATA[31:0]** Buffer Data
The peripheral's data buffer can be accessed through this 32-bit Data Port register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.9. Present State Register

**Name:** PSR
**Offset:** 0x24
**Reset:** 0x00F80000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CMDLL |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATLL[3:0] | | | | WRPPL | CARDDPL | CARDSS | CARDINS |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | BUFRDEN | BUFWREN | RTACT | WTACT |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | RTREQ | DLACT | CMDINHD | CMDINHC |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 24 – CMDLL** CMD Line Level
This status is used to check the CMD line level to recover from errors, and for debugging.

**Bits 23:20 – DATLL[3:0]** DAT[3:0] Line Level
This status is used to check the DAT line level to recover from errors, and for debugging. This is especially useful in detecting the Busy signal level from DAT[0].

**Bit 19 – WRPPL** Write Protect Pin Level
The Write Protect Switch is supported for memory and combo cards. This bit reflects the WP pin.

| Value | Description |
|---|---|
| 0 | Write protected (WP = 0) |
| 1 | Write enabled (WP = 1) |

**Bit 18 – CARDDPL** Card Detect Pin Level
This bit reflects the inverse value of the CD pin. Debouncing is not performed on this bit. This bit may be valid when CARDSS is set to 1, but it is not guaranteed because of the propagation delay. Use of this bit is limited to testing since it must be debounced by software.

| Value | Description |
|---|---|
| 0 | No card present (CD = 1) |
| 1 | Card present (CD = 0) |

**Bit 17 – CARDSS** Card State Stable
This bit is used for testing. If it is 0, the CARDDPL is not stable. If this bit is set to 1, it means that the CARDDPL is stable. No Card state can be detected if this bit is set to 1 and CARDINS is set to 0. The Software Reset For All (SWRSTALL) in SRR does not affect this bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Reset or debouncing |
| 1 | No card or card inserted |

**Bit 16 – CARDINS** Card Inserted

This bit indicates whether a card has been inserted. The peripheral debounces this signal so that the user does not need to wait for it to stabilize.
A change from 0 to 1 rises the Card Insertion (CINS) status flag in NISTR if NISTER.CINS is set to 1. An interrupt is generated if NISIER.CINS is set to 1.
A change from 1 to 0 rises the Card Removal (CREM) status flag in NISTR if NISTER.CREM is set to 1. An interrupt is generated if NISIER.CREM is set to 1.
The Software Reset For All (SWRSTALL) in SRR does not affect this bit.

**Bit 11 – BUFRDEN** Buffer Read Enable

This bit is used for non-DMA read transfers. This flag indicates that valid data exists in the peripheral data buffer. If this bit is 1, readable data exists in the buffer.
A change from 1 to 0 occurs when all the block data is read from the buffer.
A change from 0 to 1 occurs when block data is ready in the buffer. This rises the Buffer Read Ready (BRDRDY) status flag in NISTR if NISTER.BRDRDY is set to 1. An interrupt is generated if NISIER.BRDRDY is set to 1.

**Bit 10 – BUFWREN** Buffer Write Enable

This bit is used for non-DMA write transfers. This flag indicates if space is available for write data. If this bit is 1, data can be written to the buffer.
A change from 1 to 0 occurs when all the block data are written to the buffer.
A change from 0 to 1 occurs when top of block data can be written to the buffer. This rises the Buffer Write Ready (BRWRDY) status flag in NISTR if NISTER.BRWRDY is set to 1. An interrupt is generated if NISIER.BRWRDY is set to 1.

**Bit 9 – RTACT** Read Transfer Active

This bit is used to detect completion of a read transfer. Refer to section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the read command.
- When a read operation is restarted by writing a 1 to BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- When the last data block as specified by Transfer Block Size (BLKSIZE) is transferred to the system.
- In case of ADMA2, end of read is designated by the descriptor table.
- When all valid data blocks in the peripheral have been transferred to the system and no current block transfers are being sent as a result of the Stop At Block Gap Request (STPBGR) of BGCR being set to 1.

A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in NISTR if NISTER.TRFC is set to 1. An interrupt is generated if NISIER.TRFC is set to 1.

**Bit 8 – WTACT** Write Transfer Active

This bit indicates a write transfer is active. If this bit is 0, it means no valid write data exists in the peripheral. Refer to section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for more details on the sequence of events.
This bit is set to 1 in either of the following conditions:

- After the end bit of the write command.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- When a write operation is restarted by writing a 1 to BGCR.CONTR (Continue Request).

This bit is cleared to 0 in either of the following conditions:

- After getting the CRC status of the last data block as specified by the transfer count (single and multiple). In case of ADMA2, transfer count is designated by the descriptor table.
- After getting the CRC status of any block where a data transmission is about to be stopped by a Stop At Block Gap Request (STPBGR) of BGCR.

During a write transaction and as the result of the Stop At Block Gap Request (STPBGR) being set, a change from 1 to 0 rises the Block Gap Event (BLKGE) status flag in NISTR if NISTER.BLKGE is set to 1. An interrupt is generated if BLKGE is set to 1 in NISIER. This status is useful to determine whether non-DAT line commands can be issued during Write Busy.

**Bit 3 – RTREQ** Retuning Request

The peripheral can instruct the software to execute a re-tuning sequence by setting this bit when the data window is shifted by a temperature drift and a tuned sampling point does not have a good margin to receive correct data.
This bit is cleared to 0 when a command is issued by setting Execute Tuning (EXTUN) in HC2R.
A change from 0 to 1 rises the Re-Tuning Event (RTEVT) status flag in NISTR if NISTER.RTEVT is set to 1. An interrupt is generated if NISIER.RTEVT is set to 1.
This bit is not set to 1 if Sampling Clock Select (SCLKSEL) in HC2R is set to 0 (using a fixed sampling clock). Refer to Re-Tuning Modes (RTMODE) in CA1R.

| Value | Description |
|---|---|
| 0 | Fixed or well-tuned sampling clock |
| 1 | Sampling clock needs re-tuning |

**Bit 2 – DLACT** DAT Line Active

This bit indicates whether one of the DAT lines on the bus is in use.
In the case of read transactions:
This status indicates whether a read transfer is executing on the bus. A change from 1 to 0 resulting from setting the Stop At Block Gap Request (STPBGR) rises the Block Gap Event (BLKGE) status flag in NISTR if NISTER.BLKGE is set to 1. An interrupt is generated if NISIER.BLKGE is set to 1. Refer to section “Read Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
This bit is set in either of the following cases:

- After the end bit of the read command.
- When writing 1 to BGCR.CONTR (Continue Request) to restart a read transfer.

This bit is peripheral cleared in either of the following cases:

- When the end bit of the last data block is sent from the bus to the peripheral. In case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
- When a read transfer is stopped at the block gap initiated by a Stop At Block Gap Request (STPBGR).

The peripheral stops a read operation at the start of the interrupt cycle by driving the Read Wait (DAT[2] line) or by stopping the SD Clock. If the Read Wait signal is already driven (due to the fact that the data buffer cannot receive data), the peripheral can continue to stop the read operation by driving the Read Wait signal. It is necessary to support the Read Wait in order to use the Suspend/Resume operation.
In the case of write transactions:
This status indicates that a write transfer is executing on the bus. A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in NISTR if NISTER.TRFC is set to 1. An interrupt is generated if NISIER.TRFC is set to 1. Refer to section “Write Transaction Wait / Continue Timing” in the “SD Host Controller Simplified Specification V3.00” for details on timing.
This bit is set in either of the following cases:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- After the end bit of the write command.
- When writing 1 to BGCR.CONTR (Continue Request) to continue a write transfer.

This bit is cleared in either of the following cases:

- When the card releases Write Busy of the last data block. If the card does not drive a Busy signal for 8 SDCLK, the peripheral considers the card drive "Not Busy". In the case of ADMA2, the last block is designated by the last transfer of the Descriptor Table.
- When the card releases Write Busy prior to wait for write transfer as a result of a Stop At Block Gap Request (STPBGR).

Command with Busy:
This status indicates whether a command that indicates Busy (ex. erase command for memory) is executing on the bus. This bit is set to 1 after the end bit of the command with Busy and cleared when Busy is de-asserted. A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in NISTR if NISTER.TRFC is set to 1. An interrupt is generated if NISIER.TRFC is set to 1. Refer to Figures 2.11 to 2.13 in the "SD Host Controller Simplified Specification V3.00".

| Value | Description |
|---|---|
| 0 | DAT Line Inactive |
| 1 | DAT Line Active |

**Bit 1 – CMDINHD** Command Inhibit (DAT)

This status bit is 1 if either the DAT Line Active (DLACT) or the Read Transfer Active (RTACT) is set to 1. If this bit is 0, it indicates that the peripheral can issue the next command. Commands with a Busy signal belong to Command Inhibit (DAT) (ex. R1b, R5b type). A change from 1 to 0 rises the Transfer Complete (TRFC) status flag in NISTR if NISTER.TRFC is set to 1. An interrupt is generated if NISIER.TRFC is set to 1.
Note: The software can save registers in the 000-00Dh range for a suspend transaction after this bit has changed from 1 to 0.

| Value | Description |
|---|---|
| 0 | Can issue a command which uses the DAT line(s). |
| 1 | Cannot issue a command which uses the DAT line(s). |

**Bit 0 – CMDINHC** Command Inhibit (CMD)

If this bit is 0, it indicates the CMD line is not in use and the peripheral can issue a command using the CMD line. This bit is set to 1 immediately after CR is written. This bit is cleared when the command response is received. Auto CMD12 and Auto CMD23 consist of two responses. In this case, this bit is not cleared by the CMD12 or CMD23 response, but by the Read/Write command response.
Status issuing Auto CMD12 is not read from this bit. So, if a command is issued during Auto CMD12 operation, the peripheral manages to issue both commands: CMD12 and a command set by CR. Even if the Command Inhibit (DAT) is set to 1, commands using only the CMD line can be issued if this bit is 0.
A change from 1 to 0 rises the Command Complete (CMDC) status flag in NISTR if NISTER.CMDC is set to 1. An interrupt is generated if NISIER.CMDC is set to 1.
If the peripheral cannot issue the command because of a command conflict error (refer to CMDCRC in EISTR) or because of a 'Command Not Issued By Auto CMD12' error (refer to Section 1.2.31 "SDMMC Auto CMD Error Status Register"), this bit remains 1 and Command Complete is not set.

| Value | Description |
|---|---|
| 0 | Can issue a command using only CMD line. |
| 1 | Cannot issue a command. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.10. Host Control 1 Register

**Name:** HC1R
**Offset:** 0x28
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CARDDSEL | CARDDTL | | DMASEL[1:0] | | HSEN | DW | LEDCTRL |
| Access | R/W | R/W | | - | - | R/W | R/W | R/W |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – CARDDSEL** Card Detect Signal Selection

**Note:**
This register entry is specific to the SD/SDIO operation mode.

This bit selects the source for the card detection.

| Value | Description |
|---|---|
| 0 | The CD pin is selected. |
| 1 | The Card Detect Test Level (CARDDTL) is selected (for test purpose). |

**Bit 6 – CARDDTL** Card Detect Test Level

**Note:**
This register entry is specific to the SD/SDIO operation mode.

This bit is enabled while the Card Detect Signal Selection (CARDDSEL) is set to 1 and it indicates whether the card is inserted or not.

| Value | Description |
|---|---|
| 0 | No card. |
| 1 | Card inserted. |

**Bits 4:3 – DMASEL[1:0]** DMA Select

One of the supported DAM modes can be selected. The user must check support of DMA modes by referring the CA0R. Use of selected DMA is determined by DMA Enable (DMAEN) in TMR.

| Value | Name | Description |
|---|---|---|
| 0 | SDMA | SDMA is selected |
| 1 | Reserved | Reserved |
| 2 | 32BIT | 32-bit Address ADMA2 is selected |
| 3 | Reserved | Reserved |

**Bit 2 – HSEN** High Speed Enable

Before setting this bit, the user must check the High Speed Support (HSSUP) in CA0R.
If this bit is set to 0 (default), the peripheral outputs CMD line and DAT lines at the falling edge of the SD clock (up to 25 MHz). If this bit is set to 1, the SDMMC outputs the CMD line and the DAT lines at the rising edge of the SD clock (up to 50 MHz).
If Preset Value Enable (PVALEN) in HC2R is set to 1, the user needs to reset SD Clock Enable (SDCLKEN) before changing this bit to avoid generating clock glitches. After setting this bit to 1, the user sets SDCLEN to 1 again.

| Value | Description |
|---|---|
| 0 | Normal Speed mode. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | High Speed mode.<br>Note: 1. This bit is effective only if MC1R.DDR is set to 0.<br>2. The clock divider (DIV) in CCR must be set to a value different from 0 when HSEN is 1. |

**Bit 1 – DW** Data Width

This bit selects the data width of the peripheral. It must be set to match the data width of the card.
**Note:** If the Extended Data Transfer Width is 1, this bit has no effect and the data width is 8-bit mode.

| Value | Name | Description |
|---|---|---|
| 0 | 1BIT | 1-bit mode |
| 1 | 4BIT | 4-bit mode |

**Bit 0 – LEDCTRL** LED Control

**Note:**
This register entry is specific to the SD/SDIO operation mode.

This bit is used to caution the user not to remove the card while it is being accessed. If the software is going to issue multiple commands, this bit is set to 1 during all transactions.

| Value | Name | Description |
|---|---|---|
| 0 | OFF | LED off |
| 1 | ON | LED on |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.11. Power Control Register

**Name:** PCR
**Offset:** 0x29
**Reset:** 0x0E
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SDBVSEL[2:0] | | | SDBPWR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 1 | 1 | 1 | 0 |

**Bits 3:1 – SDBVSEL[2:0]** SD Bus Voltage Select
By setting this bit, the user selects the voltage level for the card. Before setting this register, the user must check the Voltage Support in CA0R. If an unsupported voltage is selected, the system does not supply the bus voltage.

| Value | Name | Description |
|---|---|---|
| 0x0-0x4 | Reserved | Reserved |
| 0x5 | 1V8 | 1.8 Volt (Typical) |
| 0x6 | 3V0 | 3.0 Volt (Typical) |
| 0x7 | 3V3 | 3.3 Volt (Typical) |

**Bit 0 – SDBPWR** SD Bus Power
This bit is automatically cleared by the peripheral if the card is removed. If this bit is cleared, the peripheral stops driving CMD and DAT[7:0] (tri-state) and drives CK to low level.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.12. Block Gap Control Register

**Name:** BGCR
**Offset:** 0x2A
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | INTBG | RWCTRL | CONTR | STPBGR |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 3 – INTBG** Interrupt at Block Gap

**Note:**
This register entry is specific to the SD/SDIO operation mode.

This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this bit should be set to 0. When the software detects an SDIO card insertion, it sets this bit according to the CCCR of the SDIO card.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLED | Interrupt detection disabled |
| 1 | ENABLED | Interrupt detection enabled |

**Bit 2 – RWCTRL** Read Wait Control

**Note:**
This register entry is specific to the SD/SDIO operation mode.

The Read Wait control is optional for SDIO cards. If the card supports Read Wait, set this bit to enable use of the Read Wait protocol to stop read data using the DAT[2] line. Otherwise, the peripheral stops the SDCLK to hold read data, which restricts command generation. When the software detects an SD card insertion, this bit must be set according to the CCCR of the SDIO card. If the card does not support Read Wait, this bit shall never be set to 1, otherwise an DAT line conflict may occur. If this bit is set to 0, Suspend/Resume cannot be supported.

| Value | Description |
|---|---|
| 0 | Disables Read Wait control. |
| 1 | Enables Read Wait control. |

**Bit 1 – CONTR** Continue Request

This bit is used to restart a transaction which was stopped using a Stop At Block Gap Request (STPBGR). To cancel stop at the block gap, set STPBGR to 0 and set this bit to 1 to restart the transfer.
The peripheral automatically clears this bit in either of the following cases:

- In the case of a read transaction, the DAT Line Active (DLACT) changes from 0 to 1 as a read transaction restarts.
- In the case of a write transaction, the Write Transfer Active (WTACT) changes from 0 to 1 as the write transaction restarts.

Therefore, it is not necessary to set this bit to 0. If STPBGR is set to 1, any write to this bit is ignored. Refer to the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

| Value | Description |
|---|---|
| 0 | No affect |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Restart |

**Bit 0 – STPBGR** Stop At Block Gap Request

This bit is used to stop executing read and write transactions at the next block gap for non-DMA, SDMA, and ADMA transfers. The user must leave this bit set to 1 until Transfer Complete (TRFC) in NISTR. Clearing both Stop At Block Gap Request and Continue Request does not cause the transaction to restart. This bit can be set whether the card supports the Read Wait signal or not. During read transfers, the peripheral stops the transaction by using the Read Wait signal (DAT[2]) if supported, or by stopping the SD clock otherwise.
In case of write transfers in which the user writes data to BDPR, this bit must be set to 1 after all the block of data is written. If this bit is set to 1, the user does not write data to BDPR.
This bit affects Read Transfer Active (RTACT), Write Transfer Active (WTACT), DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in PSR.
Refer to the “Abort Transaction” and “Suspend/Resume” sections in the “SD Host Controller Simplified Specification V3.00” for more details.

| Value | Description |
|---|---|
| 0 | Transfer |
| 1 | Stop |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.13. Wakeup Control Register: SD/SDIO

**Name:** WCR
**Offset:** 0x2B
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WKENCREM | WKENCINS | WKENCINT |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – WKENCREM** Wake-up Event Enable on Card Removal
This bit enables a wake-up event via Card Removal (CREM) in NISTR. FN_WUS (Wake-Up Support) in the CIS (Card Information Structure) does not affect this bit.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Wake-Up Event disabled |
| 1 | ENABLE | Wake-Up Event enabled |

**Bit 1 – WKENCINS** Wake-Up Event Enable on Card Insertion
This bit enables a wake-up event via Card Insertion (CINS) in NISTR. FN_WUS (Wake-Up Support) in the CIS (Card Information Structure) does not affect this bit.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Wake-Up Event disabled |
| 1 | ENABLE | Wake-Up Event enabled |

**Bit 0 – WKENCINT** Wake-Up Event Enable on Card Interrupt
This bit enables a wake-up event via Card Interrupt (CINT) in NISTR. This bit can be set to 1 if FN_WUS (Wake-Up Support) in the CIS (Card Information Structure) is set to 1 in the SDIO card.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Wake-Up Event disabled |
| 1 | ENABLE | Wake-Up Event enabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.14. Clock Control Register

**Name:** CCR
**Offset:** 0x2C
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SDCLKFSEL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | USDCLKFSEL[1:0] | | CLKGSEL | | | SDCLKEN | INTCLKS | INTCLKEN |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

**Bits 15:8 – SDCLKFSEL[7:0]** SDCLK Frequency Select

This register is used to select the frequency of the SDCLK pin. There are two SDCLK Frequency modes according to Clock Generator Select (CLKGSEL).
The length of the clock divider (DIV) is extended to 10 bits (DIV[9:8] = USDCLKFSEL, DIV[7:0] = SDCLKFSEL)
– 10-bit Divided Clock Mode (CLKGSEL = 0):
$F_{SDCLK} = F_{BASECLK}/(2 \times DIV)$

. If DIV = 0 then
$F_{SDCLK} = F_{BASECLK}$

– Programmable Clock Mode (CLKGSEL = 1):
$F_{SDCLK} = F_{MULTCLK}/(DIV+1)$

This field depends on the setting of Preset Value Enable (PVALEN) in HC2R.
If HC2R.PVALEN = 0, this field is set by the user.
If HC2R.PVALEN = 1, this field is automatically set to a value specified in one of the PVR.

**Bits 7:6 – USDCLKFSEL[1:0]** Upper Bits of SDCLK Frequency Select

These bits expand the SDCLK Frequency Select (SDCLKFSEL) to 10 bits. These two bits are assigned to bit 09-08 of the clock divider as described in SDCLKFSEL.

**Bit 5 – CLKGSEL** Clock Generator Select

This bit is used to select the clock generator mode in the SDCLK Frequency Select field. If the Programmable mode is not supported (CA1R.CLKMULT (Clock Multiplier) set to 0), then this bit cannot be written and is always read at 0.
This bit depends on the setting of Preset Value Enable (PVALEN) in HC2R.
If HC2R.PVALEN = 0, this bit is set by the user.
If HC2R.PVALEN = 1, this bit is automatically set to a value specified in one of the PVRx.

| Value | Description |
|---|---|
| 0 | Divided Clock mode (BASECLK is used to generate SDCLK). |
| 1 | Programmable Clock mode (MULTCLK is used to generate SDCLK). |

**Bit 2 – SDCLKEN** SD Clock Enable

The peripheral stops the SD Clock when writing this bit to 0. SDCLK Frequency Select (SDCLKFSEL) can be changed when this bit is 0. Then, the peripheral maintains the same clock frequency until


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

SDCLK is stopped (Stop at SDCLK=0). If Card Inserted (CARDINS) in PSR is cleared, this bit is also cleared.

| Value | Description |
|---|---|
| 0 | SD Clock disabled |
| 1 | SD Clock enabled |

**Bit 1 – INTCLKS** Internal Clock Stable

This bit is set to 1 when the SD clock is stable after setting CCR.INTCLKEN (Internal Clock Enable) to 1. The user must wait to set SD Clock Enable (SDCLKEN) until this bit is set to 1.

| Value | Description |
|---|---|
| 0 | Internal clock not ready |
| 1 | Internal clock ready |

**Bit 0 – INTCLKEN** Internal Clock Enable

This bit is set to 0 when the peripheral is not used or is awaiting a wakeup interrupt. In this case, its internal clock is stopped to reach a very low power state. Registers are still able to be read and written. The clock starts to oscillate when this bit is set to 1. Once the clock oscillation is stable, the peripheral sets Internal Clock Stable (INTCLKS) in this register to 1.
This bit does not affect card detection.

| Value | Description |
|---|---|
| 0 | The internal clock stops. |
| 1 | The internal clock oscillates. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.15. Timeout Control Register

**Name:** TCR
**Offset:** 0x2E
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DTCVAL[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 3:0 – DTCVAL[3:0]** Data Timeout Counter Value
This value determines the interval at which DAT line timeouts are detected. For more information about timeout generation, refer to Data Timeout Error (DATTEO) in EISTR. When setting this register, the user can prevent inadvertent timeout events by clearing the Data Timeout Error Status Enable (in EISTER).

$$\mathrm{TIMEOUT}_{(\mu S)} = \frac{2^{13 + \mathrm{DTCVAL}}}{F_{\mathrm{BASECLK(MHz)}}}$$

Note: DTCVAL = $F_{(Hexa)}$ is reserved.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.16. Software Reset Register

**Name:** SRR
**Offset:** 0x2F
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SWRSTDAT | SWRSTCMD | SWRSTALL |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – SWRSTDAT** Software reset for DAT line
Only part of a data circuit is reset. The DMA circuit is also reset.
The following registers and bits are cleared by this bit:

- Buffer Data Port Register BDPR: BUFDATA is cleared and initialized.
- Present State Register PSR:
  - Buffer Read Enable (BUFRDEN)
  - Buffer Write Enable (BUFWREN)
  - Read Transfer Active (RTACT)
  - Write Transfer Active (WTACT)
  - DAT Line Active (DATLL)
  - Command Inhibit - DAT (CMDINHD)
- Block Gap Control Register BGCR:
  - Continue Request (CONTR)
  - Stop At Block Gap Request (STPBGR)
- Normal Interrupt Status Register NISTR:
  - Buffer Read Ready (BRDRDY)
  - Buffer Write Ready (BWRRDY)
  - DMA Interrupt (DMAINT)
  - Block Gap Event (BLKGE)
  - Transfer Complete (TRFC)

| Value | Description |
|---|---|
| 0 | Work |
| 1 | Reset |

**Bit 1 – SWRSTCMD** Software reset for CMD line
Only part of a command circuit is reset.
The following registers and bits are cleared by this bit:

- Present State Register PSR:
  - Command Inhibit (CMD) (CMDINHC)
- Normal Interrupt Status Register NISTR:
  - Command Complete (CMDC)

| Value | Description |
|---|---|
| 0 | Work |
| 1 | Reset |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – SWRSTALL** Software reset for All

This reset affects the entire peripheral except the card detection circuit. During initialization, the peripheral must be reset by setting this bit to 1. This bit is automatically cleared to 0 when CA0R and CA1R are valid and the user can read them. If this bit is set to 1, the user should issue a reset command and reinitialize the card.
List of registers cleared to 0:

- SDMA System Address / Argument 2 Register SSAR
- Block Size Register BSR
- Block Count Register BCR
- Argument 1 Register ARG1R
- Transfer Mode Register TMR
- Command Register CR
- Response Register n RR
- Buffer Data Port Register BDPR
- Present State Register PSR (except CMDLL, DATLL, WRPPL, CARDDDPL, CARDSS, CARDINS)
- Host Control 1 Register HC1R
- Power Control Register PCR
- Block Gap Control Register BGCR
- Wakeup Control Register WCR
- Clock Control Register CCR
- Timeout Control Register TCR
- Normal Interrupt Status Register NISTR
- Error Interrupt Status Register EISTR
- Normal Interrupt Status Enable Register NISTER
- Error Interrupt Status Enable Register EISTER
- Normal Interrupt Signal Enable Register NISIER
- Error Interrupt Signal Enable Register EISIER
- Auto CMD Error Status Register ACESR
- Host Control 2 Register HC2R - DEFAULT
- ADMA Error Status Register AESR
- ADMA System Address Registers ASAR
- Slot Interrupt Status Register SISR
- e.MMC Control 1 Register MC1R
- e.MMC Control 2 Register MC2R
- AHB Control Register ACR
- Clock Control 2 Register CC2R
- Capabilities Control Register CACR (except KEY)

| Value | Description |
|---|---|
| 0 | Work |
| 1 | Reset |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.17. Normal Interrupt Status Register

**Name:** NISTR
**Offset:** 0x30
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ERRINT | BOOTAR | | | | | | CINT |
| Access | R/W | R/W | | | | | | R/W |
| Reset | 0 | 0 | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CREM | CINS | BRDRDY | BWRRDY | DMAINT | BLKGE | TRFC | CMDC |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – ERRINT** Error Interrupt

If any of the bits in EISTR are set, then this bit is set. Therefore, the user can efficiently test for an error by checking this bit first. This bit is read-only.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 14 – BOOTAR** Boot Acknowledge Received

**Note:** This register entry is specific to the e.MMC operation mode.

This bit is set to 1 when the peripheral received a Boot Acknowledge pattern from the e.MMC. This bit can only be set to 1 if NISTER.BOOTAR is set to 1. An interrupt can only be generated if NISIER.BOOTAR is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | Boot Acknowledge pattern not received. |
| 1 | Boot Acknowledge pattern received. |

**Bit 8 – CINT** Card Interrupt

**Note:**
This register entry is specific to the SD/SDIO operation mode.

Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor. In 1-bit mode, the peripheral detects the Card Interrupt without SDCLK to support wake-up. In 4-bit mode, the Card Interrupt signal is sampled during the interrupt cycle, so there are some sample delays between the interrupt signal from the SD card and the interrupt to the system.
When this bit has been set to 1 and the user needs to start this interrupt service, Card Interrupt Status Enable (CINT) in NISTER may be set to 0 in order to clear the card interrupt statuses latched in the peripheral and to stop driving the interrupt signal to the system. After completion of the card interrupt service (it should reset interrupt factors in the SD card and the interrupt signal may not be asserted), set NISTER.CINT to 1 and start sampling the interrupt signal again.
Interrupt detected by DAT[1] is supported when there is one card per slot. In case of a shared bus, interrupt pins are used to detect interrupts. If 0 is set to Interrupt Pin Select (INTPSEL) in SBCR, this status is effective. If a non-zero value is set to INTPSEL, INT_A, INT_B or INT_C is used as device interrupts.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit can only be set to 1 if NISTER.CREM is set to 1. An interrupt can only be generated if NISIER.CREM is set to 1.

| Value | Description |
|---|---|
| 0 | No card interrupt |
| 1 | Card interrupt |

**Bit 7 – CREM** Card Removal

**Note:**
This register entry is specific to the SD/SDIO operation mode.

This status is set to 1 if Card Inserted (CARDINS) in PSR changes from 1 to 0. When the user writes this bit to 1 to clear this status, the status of PSR.CARDINS must be confirmed because the card detect state may possibly be changed when the user clears this bit and no interrupt event can be generated.
This bit can only be set to 1 if NISTER.CREM is set to 1. An interrupt can only be generated if NISIER.CREM is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | Card state unstable or card inserted |
| 1 | Card removed |

**Bit 6 – CINS** Card Insertion

**Note:**
This register entry is specific to the SD/SDIO operation mode.

This status is set if Card Inserted (CARDINS) in PSR changes from 0 to 1. When the user writes this bit to 1 to clear this status, the status of PSR.CARDINS must be confirmed because the card detect state may possibly be changed when the user clears this bit and no interrupt event can be generated.
This bit can only be set to 1 if NISTER.CINS is set to 1. An interrupt can only be generated if NISIER.CINS is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | Card state unstable or card removed |
| 1 | Card inserted |

**Bit 5 – BRDRDY** Buffer Read Ready

This status is set to 1 if the Buffer Read Enable (BUFRDEN) changes from 0 to 1. Refer to BUFRDEN in PSR.
This bit can only be set to 1 if NISTER.BRDRDY is set to 1. An interrupt can only be generated if NISIER.BRDRDY is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | Not ready to read buffer |
| 1 | Ready to read buffer |

**Bit 4 – BWRRDY** Buffer Write Ready

This status is set to 1 if the Buffer Write Enable (BUFWREN) changes from 0 to 1. Refer to BUFWREN in PSR.
This bit can only be set to 1 if NISTER.BWRRDY is set to 1. An interrupt can only be generated if NISIER.BWRRDY is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | Not ready to write buffer |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Ready to write buffer |

**Bit 3 – DMAINT** DMA Interrupt

This status is set if the peripheral detects the Host SDMA Buffer boundary during transfer. Refer to SDMA Buffer Boundary (BOUNDARY) in BSR.
In case of ADMA, by setting the "int" field in the descriptor table, the peripheral rises this status flag when the descriptor line is completed. This status flag does not rise after Transfer Complete (TRFC).
This bit can only be set to 1 if NISTER.DMAINT is set to 1. An interrupt can only be generated if NISIER.DMAINT is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No DMA Interrupt |
| 1 | DMA Interrupt |

**Bit 2 – BLKGE** Block Gap Event

If the Stop At Block Gap Request (STPBGR) in BGCR is set to 1, this bit is set when either a read or a write transaction is stopped at a block gap. If STPBGR is not set to 1, this bit is not set to 1.
In the case of a Read transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status (when the transaction is stopped at SD bus timing). The Read Wait must be supported in order to use this function. Refer to section "Read Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" about the detailed timing.
In the case of a Write transaction:
This bit is set at the falling edge of the Write Transfer Active (WTACT) status (after getting the CRC status at SD bus timing). Refer to section "Write Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
This bit can only be set to 1 if NISTER.BLKGE is set to 1. An interrupt can only be generated if NISIER.BLKGE is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No block gap event |
| 1 | Transaction stopped at block gap |

**Bit 1 – TRFC** Transfer Complete

This bit is set when a read/write transfer and a command with Busy is completed.
In the case of a Read Transaction:
This bit is set at the falling edge of the Read Transfer Active Status. The interrupt is generated in two cases. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request (STPBGR) in BGCR (after valid data has been read to the system). Refer to section "Read Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
In the case of a Write Transaction:
This bit is set at the falling edge of the DAT Line Active (DLACT) status. This interrupt is generated in two cases. The first is when the last data is written to the card as specified by the data length and the Busy signal is released. The second is when data transfers are stopped at the block gap by setting Stop At Block Gap Request (STPBGR) in BGCR and data transfers are completed. (After valid data is written to the card and the Busy signal is released). Refer to section "Write Transaction Wait / Continue Timing" in the "SD Host Controller Simplified Specification V3.00" for more details on the sequence of events.
In the case of command with Busy:
This bit is set when Busy is de-asserted. Refer to DAT Line Active (DLACT) and Command Inhibit (DAT) (CMDINHD) in PSR.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit can only be set to 1 if NISTER.TRFC is set to 1. An interrupt can only be generated if NISIER.TRFC is set to 1.
Writing this bit to 1 clears this bit.
The table below shows that Transfer Complete (TRFC) has a higher priority than Data Timeout Error (DATTEO). If both bits are set to 1, execution of a command can be considered to be completed.

| TRFC | DATTEO | Meaning of the status |
|---|---|---|
| 0 | 0 | Interrupted by another factor |
| 0 | 1 | Timeout occurred during transfer |
| 1 | Don't Care | Command execution complete |

| Value | Description |
|---|---|
| 0 | Command execution is not complete. |
| 1 | Command execution is complete. |

**Bit 0 – CMDC** Command Complete

This bit is set when getting the end bit of the command response. Auto CMD12 and Auto CMD23 consist of two responses. Command Complete is not generated by the response of CMD12 or CMD23, but it is generated by the response of a read/write command. Refer to Command Inhibit (CMD) in PSR for details on how to control this bit.
This bit can only be set to 1 if NISTER.CMDC is set to 1. An interrupt can only be generated if NISIER.CMDC is set to 1.
Writing this bit to 1 clears this bit.
The table below shows that Command Timeout Error (CMDTEO) has a higher priority than Command Complete (CMDC). If both bits are set to 1, it can be considered that the response was not received correctly.

| CMDC | CMDTEO | Meaning of the status |
|---|---|---|
| 0 | 0 | Interrupted by another factor |
| Don't care | 1 | Response not received within 64 SDCLK cycles |
| 1 | 0 | Response received |

| Value | Description |
|---|---|
| 0 | No command complete |
| 1 | Command complete |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.18. Error Interrupt Status Register

**Name:** EISTR
**Offset:** 0x32
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | BOOTAE | | | ADMA | ACMD |
| Access | | | | R/W | | | R/W | R/W |
| Reset | | | | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 12 – BOOTAE** Boot Acknowledge Error

**Note:** This register entry is specific to the e.MMC operation mode.

This bit is set to 1 when detecting that the e.MMC Boot Acknowledge Status has a value other than "010".
This bit can only be set to 1 if EISTER.BOOTAE is set to 1. An interrupt can only be generated if EISIER.BOOTAE is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 9 – ADMA** ADMA Error

This bit is set to 1 when the peripheral detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in AESR.
In addition, the peripheral rises this status bit when it detects some invalid description data (Valid=0) at the ST_FDS state (refer to section "Advanced DMA" in the "SD Host Controller Simplified Specification V3.00". ADMA Error Status (ERRST) in AESR indicates that an error occurs in ST_FDS state. The user may find that the Valid bit is not set at the error descriptor.
This bit can only be set to 1 if EISTER.ADMA is set to 1. An interrupt can only be generated if EISIER.ADMA is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 8 – ACMD** Auto CMD Error

Auto CMD12 and Auto CMD23 use this error status. This bit is set to 1 when detecting that one of the 0 to 4 bits in AESR (ACESR[4:0]) has changed from 0 to 1. In the case of Auto CMD12, this bit is set to 1, not only when errors occur in Auto CMD12, but also when Auto CMD12 is not executed due to the previous command error.
This bit can only be set to 1 if EISTER.ACMD is set to 1. An interrupt can only be generated if EISIER.ACMD is set to 1.
Writing this bit to 1 clears this bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 7 – CURLIM** Current Limit Error

By setting SD Bus Power (SDBPWR) in PCR, the peripheral is requested to supply power for the SD Bus. The peripheral is protected from an illegal card by stopping power supply to the card, in which case this bit indicates a failure status. Reading 1 means the peripheral is not supplying power to the card due to some failure. Reading 0 means that the peripheral is supplying power and no error has occurred. The peripheral may require some sampling time to detect the current limit.
This bit can only be set to 1 if EISTER.CURLIM is set to 1. An interrupt can only be generated if EISIER.CURLIM is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 6 – DATEND** Data End Bit Error

This bit is set to 1 either when detecting 0 at the end bit position of read data which uses the DAT line or at the end bit position of the CRC Status.
This bit can only be set to 1 if EISTER.DATEND is set to 1. An interrupt can only be generated if EISIER.DATEND is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 5 – DATCRC** Data CRC Error

This bit is set to 1 when detecting a CRC error during a transfer of read data which uses the DAT line or when detecting that the Write CRC Status has a value other than '010'.
This bit can only be set to 1 if EISTER. DATCRC is set to 1. An interrupt can only be generated if EISIER. DATCRC is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 4 – DATTEO** Data Timeout error

This bit is set to 1 when detecting one of following timeout conditions:

- Busy timeout for R1b, R5b response type (see “Physical Layer Simplified Specification V3.01” and “SDIO Simplified Specification V3.00” ).
- Busy timeout after Write CRC Status.
- Write CRC Status timeout.
- Read data timeout.

This bit can only be set to 1 if EISTER.DATTEO is set to 1. An interrupt can only be generated if EISIER.DATTEO is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 3 – CMDIDX** Command Index Error

This bit is set to 1 if a Command Index error occurs in the command response.
This bit can only be set to 1 if EISTER.CMDIDX is set to 1. An interrupt can only be generated if EISIER.CMDIDX is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 2 – CMDEND** Command End Bit Error

This bit is set to 1 when detecting that the end bit of a command response is 0.
This bit can only be set to 1 if EISTER.CMDEND is set to 1. An interrupt can only be generated if EISIER.CMDEND is set to 1.
Writing this bit to 1 clears this bit.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 1 – CMDCRC** Command CRC Error

The Command CRC Error is generated in two cases.
If a response is returned and Command Timeout Error (CMDTEO) is set to 0 (indicating no command timeout), this bit is set to 1 when detecting a CRC error in the command response.
The peripheral detects a CMD line conflict by monitoring the CMD line when a command is issued. If the peripheral drives the CMD line to 1 level, but detects 0 level on the CMD line at the next SDCLK edge, then the peripheral aborts the command (stops driving the CMD line) and sets this bit to 1. CMDTEO is also set to 1 to indicate a CMD line conflict (refer to Table 40-2).
This bit can only be set to 1 if EISTER.CMDCRC is set to 1. An interrupt can only be generated if EISIER.CMDCRC is set to 1.
Writing this bit to 1 clears this bit.

**Bit 0 – CMDTEO** Command Timeout Error

This bit is set to 1 only if no response is returned within 64 SDCLK cycles from the end bit of the command. If the peripheral detects a CMD line conflict, in which case Command CRC Error (CMDCRC) is also set to 1 (refer to Table 40-2), this bit is set without waiting for 64 SDCLK cycles because the command is aborted by the peripheral.
This bit can only be set to 1 if EISTER.CMDTEO is set to 1. An interrupt can only be generated if EISIER.CMDTEO is set to 1.
Writing this bit to 1 clears this bit.

**Table 40-2.** CMD Error Types

| CMDCRC | CMDTEO | Types of error |
|---|---|---|
| 0 | 0 | No error |
| 0 | 1 | Response timeout error |
| 1 | 0 | Response CRC error |
| 1 | 1 | CMD line conflict |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.19. Normal Interrupt Status Enable Register: e.MMC

**Name:** NISTER
**Offset:** 0x34
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | BOOTAR | | | | | | CINT |
| Access | | R/W | | | | | | R/W |
| Reset | | 0 | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CREM | CINS | BRDRDY | BWRRDY | DMAINT | BLKGE | TRFC | CMDC |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 14 – BOOTAR** Boot Acknowledge Received Status Enable
**Note:** This register entry is specific to the e.MMC operation mode.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The BOOTAR status flag in NISTR is masked. |
| 1 | ENABLED | The BOOTAR status flag in NISTR is enabled. |

**Bit 8 – CINT** Card Interrupt Status Enable
If this bit is set to 0, the peripheral clears interrupt requests to the system. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. The user may clear this bit before servicing the Card Interrupt and may set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CINT status flag in NISTR is masked. |
| 1 | ENABLED | The CINT status flag in NISTR is enabled. |

**Bit 7 – CREM** Card Removal Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CREM status flag in NISTR is masked. |
| 1 | ENABLED | The CREM status flag in NISTR is enabled. |

**Bit 6 – CINS** Card Insertion Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CINS status flag in NISTR is masked. |
| 1 | ENABLED | The CINS status flag in NISTR is enabled. |

**Bit 5 – BRDRDY** Buffer Read Ready Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The BRDRDY status flag in NISTR is masked. |
| 1 | ENABLED | The BRDRDY status flag in NISTR is enabled. |

**Bit 4 – BWRRDY** Buffer Write Ready Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The BWRRDY status flag in NISTR is masked. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 1 | ENABLED | The BWRRDY status flag in NISTR is enabled. |

**Bit 3 – DMAINT** DMA Interrupt Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The DMAINT status flag in NISTR is masked. |
| 1 | ENABLED | The DMAINT status flag in NISTR is enabled. |

**Bit 2 – BLKGE** Block Gap Event Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The BLKGE status flag in NISTR is masked. |
| 1 | ENABLED | The BLKGE status flag in NISTR is enabled. |

**Bit 1 – TRFC** Transfer Complete Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The TRFC status flag in NISTR is masked. |
| 1 | ENABLED | The TRFC status flag in NISTR is enabled. |

**Bit 0 – CMDC** Command Complete Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CMDC status flag in NISTR is masked. |
| 1 | ENABLED | The CMDC status flag in NISTR is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.20. Error Interrupt Status Enable Register

**Name:** EISTER
**Offset:** 0x36
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | BOOTAE | | | ADMA | ACMD |
| Access | | | | R/W | | | R/W | R/W |
| Reset | | | | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 12 – BOOTAE** Boot Acknowledge Error Status Enable
**Note:** This register entry is specific to the e.MMC operation mode.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The BOOTAE status flag in EISTR is masked. |
| 1 | ENABLED | The BOOTAE status flag in EISTR is enabled. |

**Bit 9 – ADMA** ADMA Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The ADMA status flag in EISTR is masked. |
| 1 | ENABLED | The ADMA status flag in EISTR is enabled. |

**Bit 8 – ACMD** Auto CMD Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The ACMD status flag in EISTR is masked. |
| 1 | ENABLED | The ACMD status flag in EISTR is enabled. |

**Bit 7 – CURLIM** Current Limit Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CURLIM status flag in EISTR is masked. |
| 1 | ENABLED | The CURLIM status flag in EISTR is enabled. |

**Bit 6 – DATEND** Data End Bit Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The DATEND status flag in EISTR is masked. |
| 1 | ENABLED | The DATEND status flag in EISTR is enabled. |

**Bit 5 – DATCRC** Data CRC Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The DATCRC status flag in EISTR is masked. |
| 1 | ENABLED | The DATCRC status flag in EISTR is enabled. |

**Bit 4 – DATTEO** Data Timeout Error Status Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The DATTEO status flag in EISTR is masked. |
| 1 | ENABLED | The DATTEO status flag in EISTR is enabled. |

**Bit 3 – CMDIDX** Command Index Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CMDIDX status flag in EISTR is masked. |
| 1 | ENABLED | The CMDIDX status flag in EISTR is enabled. |

**Bit 2 – CMDEND** Command End Bit Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CMDEND status flag in EISTR is masked. |
| 1 | ENABLED | The CMDEND status flag in EISTR is enabled. |

**Bit 1 – CMDCRC** Command CRC Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CMDCRC status flag in EISTR is masked. |
| 1 | ENABLED | The CMDCRC status flag in EISTR is enabled. |

**Bit 0 – CMDTEO** Command Timeout Error Status Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | The CMDTEO status flag in EISTR is masked. |
| 1 | ENABLED | The CMDTEO status flag in EISTR is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 40.8.21. Normal Interrupt Signal Enable Register

**Name:** NISIER
**Offset:** 0x38
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | BOOTAR | | | | | | CINT |
| Access | | R/W | | | | | | R/W |
| Reset | | 0 | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CREM | CINS | BRDRDY | BWRRDY | DMAINT | BLKGE | TRFC | CMDC |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 14 – BOOTAR** Boot Acknowledge Received Signal Enable
**Note:** This register entry is specific to the e.MMC operation mode.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.BOOTAR is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.BOOTAR is set. |

**Bit 8 – CINT** Card Interrupt Signal Enable
**Note:**
This register entry is specific to the SD/SDIO operation mode.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.CINT is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.CINT is set. |

**Bit 7 – CREM** Card Removal Signal Enable
**Note:**
This register entry is specific to the SD/SDIO operation mode.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.CREM is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.CREM is set. |

**Bit 6 – CINS** Card Insertion Signal Enable
**Note:**
This register entry is specific to the SD/SDIO operation mode.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.CINS is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.CINS is set. |

**Bit 5 – BRDRDY** Buffer Read Ready Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.BRDRDY is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.BRDRDY is set. |

**Bit 4 – BWRRDY** Buffer Write Ready Signal Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.BWRRDY is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.BWRRDY is set. |

**Bit 3 – DMAINT** DMA Interrupt Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.DMAINT is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.DMAINT is set. |

**Bit 2 – BLKGE** Block Gap Event Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.BLKGE is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.BLKGE is set. |

**Bit 1 – TRFC** Transfer Complete Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.TRFC is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.TRFC is set. |

**Bit 0 – CMDC** Command Complete Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when NISTR.CMDC is set. |
| 1 | ENABLED | An interrupt is generated when NISTR.CMDC is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.22. Error Interrupt Signal Enable Register

**Name:** EISIER
**Offset:** 0x3A
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | BOOTAE | | | ADMA | ACMD |
| Access | | | | R/W | | | R/W | R/W |
| Reset | | | | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 12 – BOOTAE** Boot Acknowledge Error Signal Enable
**Note:** This register entry is specific to the e.MMC operation mode.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.BOOTAE is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.BOOTAE is set. |

**Bit 9 – ADMA** ADMA Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.ADMA is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.ADMA is set. |

**Bit 8 – ACMD** Auto CMD Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.ACMD is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.ACMD is set. |

**Bit 7 – CURLIM** Current Limit Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.CURLIM is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.CURLIM is set. |

**Bit 6 – DATEND** Data End Bit Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.DATEND is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.DATEND is set. |

**Bit 5 – DATCRC** Data CRC Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.DATCRC is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.DATCRC is set. |

**Bit 4 – DATTEO** Data Timeout Error Signal Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.DATTEO is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.DATTEO is set. |

**Bit 3 – CMDIDX** Command Index Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.CMDIDX is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.CMDIDX is set. |

**Bit 2 – CMDEND** Command End Bit Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.CMDEND is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.CMDEND is set. |

**Bit 1 – CMDCRC** Command CRC Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.CMDCRC is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.CMDCRC is set. |

**Bit 0 – CMDTEO** Command Timeout Error Signal Enable

| Value | Name | Description |
|---|---|---|
| 0 | MASKED | No interrupt is generated when EISTR.CMDTEO is set. |
| 1 | ENABLED | An interrupt is generated when EISTR.CMDTEO is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.23. Auto CMD Error Status Register

**Name:** ACESR
**Offset:** 0x3C
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMDNI | | | ACMDIDX | ACMDEND | ACMDCRC | ACMDTEO | ACMD12NE |
| Access | R | | | R | R | R | R | R |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – CMDNI** Command Not Issued by Auto CMD12 Error
Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (ACESR[4:1]). This bit is set to 0 when Auto CMD Error is generated by Auto CMD23.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 4 – ACMDIDX** Auto CMD Index Error
This bit is set to 1 when the Command Index error occurs in response to a command.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 3 – ACMDEND** Auto CMD End Bit Error
This bit is set to 1 when detecting that the end bit of the command response is 0.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bit 2 – ACMDCRC** Auto CMD CRC Error
This bit is set to 1 when detecting a CRC error in the command response (refer to Table 1-7 for more details).

**Bit 1 – ACMDTEO** Auto CMD Timeout Error
This bit is set to 1 if no response is returned within 64 SDCLK cycles from the end bit of the command. If this bit is set to 1, the other error status bits (ACESR[4:2]) are meaningless.

| ACMDCRC | ACMDTEO | Types of error |
|---|---|---|
| 0 | 0 | No error |
| 0 | 1 | Response Timeout error |
| 1 | 0 | Response CRC error |
| 1 | 1 | CMD line conflict |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – ACMD12NE** Auto CMD12 Not Executed

If a memory multiple block data transfer is not started due to a command error, this bit is not set to 1 because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the peripheral cannot issue Auto CMD12 to stop a memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (ACESR[4:1]) are meaningless.
This bit is set to 0 when an Auto CMD error is generated by Auto CMD23.

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 40.8.24. Host Control 2 Register: e.MMC

**Name:** HC2R - EMMC
**Offset:** 0x3E
**Reset:** 0x0000
**Property:** -

**Note:** The content of the HC2R register is depending on the mode. This description is for e.MMC mode. For SD/SDIO mode, see HC2R - DEFAULT.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PVALEN | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SCLKSEL | EXTUN | DRVSEL[1:0] | | HS200EN[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – PVALEN** Preset Value Enable

As the operating SDCLK frequency depends on the system implementation, it is difficult to determine these parameters in the standard host driver. When Preset Value Enable (PVALEN) is set to 1, automatic SDCLK frequency generationis performed without considering system-specific conditions. This bit enables the functions defined in PVR.
If this bit is written to 0, the Clock Generator Select bit (CCR.CLKGSEL) and the SDCLK Frequency Select bit (CCR.SDCLKFSEL) in the Clock Control Register (CCR) are selected by the user.
If this bit is set to 1, CCR.SDCLKFSEL and .CLKGSEL and HC2R.DRVSEL are set by the peripheral as specified in the Preset Value Register (PVR).

| Value | Description |
|---|---|
| 0 | CCR.SDCLK, CCR.SDCLKFSEL controlled by the user. |
| 1 | Automatic selection by Preset Value is enabled. |

**Bit 7 – SCLKSEL** Sampling Clock Select

The peripheral uses this bit to select the sampling clock to receive CMD and DAT.
This bit is set by the tuning procedure and is valid after completion of tuning (when EXTUN is cleared). Setting 1 means that tuning is completed successfully and setting 0 means that tuning has failed.
Writing 1 to this bit is meaningless and ignored. A tuning circuit is reset by writing to 0. This bit can be cleared by setting EXTUN to 1. Once the tuning circuit is reset, it takes time to complete a tuning sequence. Therefore, the user should keep this bit to 1 to perform a re-tuning sequence to complete a re-tuning sequence in a short time. Changing this bit is not allowed while the peripheral is receiving a response or a read data block. Refer to Figure 2.29 in the “SD Host Controller Simplified Specification V3.00” .

| Value | Description |
|---|---|
| 0 | The fixed clock is used to sample data. |
| 1 | The tuned clock is used to sample data. |

**Bit 6 – EXTUN** Execute Tuning

This bit is set to 1 to start the tuning procedure and is automatically cleared when the tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select (SCLKSEL). The tuning procedure is aborted by writing 0. Refer to Figure 2.29 in the “SD Host Controller Simplified Specification V3.00” .


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Not tuned or tuning completed |
| 1 | Execute tuning |

**Bits 5:4 – DRVSEL[1:0]** Driver Strength Select
The peripheral output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set according to the Driver Type A, C and D support bits in CA1R.
This field depends on setting of Preset Value Enable (PVALEN):

- PVALEN=0 - This field is set by the user.
- PVALEN=1 - This field is automatically set by a value specified in one of the PVRx.

| Value | Name | Description |
|---|---|---|
| 0 | TYPEB | Driver Type B is selected (Default) |
| 1 | TYPEA | Driver Type A is selected |
| 2 | TYPEC | Driver Type C is selected |
| 3 | TYPED | Driver Type D is selected |

**Bits 3:0 – HS200EN[3:0]** HS200 Mode Enable
This field is used to select the e.MMC HS200 mode. When HS200EN is set to $B_{(hexa)}$, the HS200 mode is enabled. Any other value except 0 is forbidden when interfacing an e.MMC device.
If Preset Value Enable is set to 1, peripheral sets SDCLK Frequency Select (SDCLKFSEL), Clock Generator Select (CLKGSEL) in CCR and Driver Strength Select (DRVSEL) according to PVR. In this case, one of the preset value registers is selected by this field. The user needs to reset SD Clock Enable (SDCLKEN) before changing this field to avoid generating a clock glitch. After setting this field, the user sets SDCLKEN to 1 again.
**Note:** This field is effective only if MC1R.DDR is written to 0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.25. Host Control 2 Register: SD/SDIO

**Name:** HC2R - DEFAULT
**Offset:** 0x3E
**Reset:** 0x0000
**Property:** -

**Note:** The content of the HC2R register is depending on the mode. This description is for SD/SDIO mode. For e.MMC mode, see HC2R - EMMC.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PVALEN | ASINTEN | | | | | | |
| Access | R/W | R/W | | | | | | |
| Reset | 0 | 0 | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SCLKSEL | EXTUN | DRVSEL[1:0] | | VS18EN | UHSMS[2:0] | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – PVALEN** Preset Value Enable

As the operating SDCLK frequency depends on the system implementation, it is difficult to determine these parameters in the standard host driver. When Preset Value Enable (PVALEN) is set to 1, automatic SDCLK frequency generationis performed without considering system-specific conditions. This bit enables the functions defined in PVR.
If this bit is written to 0, the Clock Generator Select bit (CCR.CLKGSEL) and the SDCLK Frequency Select bit (CCR.SDCLKFSEL) in the Clock Control Register (CCR) are selected by the user.
If this bit is set to 1, CCR.SDCLKFSEL and .CLKGSEL and HC2R.DRVSEL are set by the peripheral as specified in the Preset Value Register (PVR).

| Value | Description |
|---|---|
| 0 | CCR.SDCLK, CCR.SDCLKFSEL controlled by the user. |
| 1 | Automatic selection by Preset Value is enabled. |

**Bit 14 – ASINTEN** Asynchronous Interrupt Enable

This bit can be set to 1 if a card support asynchronous interrupts and Asynchronous Interrupt Support (ASINTSUP) is set to 1 in CA0R. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode. If this bit is set to 1, the user can stop the SDCLK during the asynchronous interrupt period to save power. During this period, the peripheral continues to deliver the Card Interrupt to the host when it is asserted by the card.

| Value | Description |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

**Bit 7 – SCLKSEL** Sampling Clock Select

The peripheral uses this bit to select the sampling clock to receive CMD and DAT.
This bit is set by the tuning procedure and is valid after completion of tuning (when EXTUN is cleared). Setting 1 means that tuning is completed successfully and setting 0 means that tuning has failed.
Writing 1 to this bit is meaningless and ignored. A tuning circuit is reset by writing to 0. This bit can be cleared by setting EXTUN to 1. Once the tuning circuit is reset, it takes time to complete the tuning sequence. Therefore, the user should keep this bit to 1 to perform a re-tuning sequence to complete a re-tuning sequence in a short time. Changing this bit is not allowed while the peripheral


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

is receiving a response or a read data block. Refer to Figure 2.29 in the “SD Host Controller Simplified Specification V3.00” .

| Value | Description |
|---|---|
| 0 | The fixed clock is used to sample data. |
| 1 | The tuned clock is used to sample data. |

**Bit 6 – EXTUN** Execute Tuning

This bit is set to 1 to start the tuning procedure and is automatically cleared when the tuning procedure is completed. The result of tuning is indicated to Sampling Clock Select (SCLKSEL). The tuning procedure is aborted by writing 0. Refer to Figure 2.29 in the “SD Host Controller Simplified Specification V3.00” .

| Value | Description |
|---|---|
| 0 | Not tuned or tuning completed |
| 1 | Execute tuning |

**Bits 5:4 – DRVSEL[1:0]** Driver Strength Select

The peripheral output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set according to the Driver Type A, C and D support bits in CA1R.
This field depends on setting of Preset Value Enable (PVALEN):

- PVALEN=0 - This field is set by the user.
- PVALEN=1 - This field is automatically set by a value specified in one of the PVRx.

| Value | Name | Description |
|---|---|---|
| 0 | TYPEB | Driver Type B is selected (Default) |
| 1 | TYPEA | Driver Type A is selected |
| 2 | TYPEC | Driver Type C is selected |
| 3 | TYPED | Driver Type D is selected |

**Bit 3 – VS18EN** 1.8V Signaling Enable

This bit controls the voltage regulator for the I/O cell. 3.3V is supplied to the card regardless of the signaling voltage.
Setting this bit from 0 to 1 starts changing the signal voltage from 3.3V to 1.8V. The 1.8V regulator output must be stable within 5 ms.
Clearing this bit from 1 to 0 starts changing the signal voltage from 1.8V to 3.3V. The 3.3V regulator output must be stable within 5ms.
The user can set this bit to 1 when the peripheral supports 1.8V signaling (one of the support bits is set to 1: SDR50SUP, SDR104SUP or DDR50SUP in CA1R) and the card or device supports UHS-I (S18A = 1. Refer to “Bus Switch Voltage Switch Sequence in the “Physical Layer Simplified Specification V3.01” ).

| Value | Description |
|---|---|
| 0 | 3.3V signaling |
| 1 | 1.8V signaling |

**Bits 2:0 – UHSMS[2:0]** UHS Mode Select

This field is used to select one of the UHS-I modes and is effective when 1.8V Signal Enable (VS18EN) is set to 1.
If Preset Value Enable is set to 1, the peripheral sets SDCLK Frequency Select (SDCLKFSEL), Clock Generator Select (CLKGSEL) in CCR and Driver Strength Select (DRVSEL) according to PVR. In this case, one of the preset value registers is selected by this field. The user needs to reset SD Clock Enable (SDCLKEN) before changing this field to avoid generating a clock glitch. After setting this field, the user sets SDCLKEN to 1 again.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0 | SDR12 | UHS SDR12 Mode |
| 1 | SDR25 | UHS SDR25 Mode |
| 2 | SDR50 | UHS SDR50 Mode |
| 3 | SDR104 | UHS SDR104 Mode |
| 4 | DDR50 | UHS DDR50 Mode<br>Note: This field is effective only if MC1R.DDR is set to 0. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 40.8.26. Capabilities 0 Register

**Name:** CA0R
**Offset:** 0x40
**Reset:** 0x27E80080
**Property:** -

**Note:** The Capabilities 0 Register is not supposed to be written by the user.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SLTYPE[1:0] | | ASINTSUP | SB64SUP | | V18VSUP | V30VSUP | V33VSUP |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 1 | 0 | | 1 | 1 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SRSUP | SDMASUP | HSSUP | | ADMA2SUP | ED8SUP | MAXBLKL[1:0] | |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | | 1 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BASECLKF[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TEOCLKU | | TEOCLKF[5:0] | | | | | |
| Access | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:30 – SLTYPE[1:0]** Slot Type
This field indicates usage of a slot by a specific system. A Peripheral Control register set is defined per slot.
Embedded Slot for One Device means that only one non-removable device is connected to a bus slot.
The Standard Host Driver controls a removable card (SLTYPE = 0) or one embedded device (SLTYPE = 1) connected to an SD bus slot.

| Value | Name |
|---|---|
| 0 | Removable Card Slot |
| 1 | Embedded Slot for One Device |
| 2 | Shared Bus Slot |
| 2 | Reserved |
| 3 | Reserved |

**Bit 29 – ASINTSUP** Asynchronous Interrupt Support
Refer to the section “Asynchronous Interrupt” in the “SDIO Simplified Specification V3.00”.

| Value | Description |
|---|---|
| 0 | Asynchronous interrupt not supported |
| 1 | Asynchronous interrupt supported |

**Bit 28 – SB64SUP** 64-Bit System Bus Support
This bit indicates if the peripheral supports the 64-bit Address Descriptor mode and is connected to the 64-bit address system bus.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | 64-bit address bus not supported |
| 1 | 64-bit address bus supported |

**Bit 26 – V18VSUP** Voltage Support 1.8V

| Value | Description |
|---|---|
| 0 | 1.8V voltage supply not supported |
| 1 | 1.8V voltage supply supported |

**Bit 25 – V30VSUP** Voltage Support 3.0V

**Note:** The signal and supply voltages of the peripheral are limited by the supply voltage of the device.

| Value | Description |
|---|---|
| 0 | 3.0V voltage supply not supported |
| 1 | 3.0V voltage supply supported |

**Bit 24 – V33VSUP** Voltage Support 3.3V

**Note:** The signal and supply voltages of the peripheral are limited by the supply voltage of the device.

| Value | Description |
|---|---|
| 0 | 3.3V voltage supply not supported |
| 1 | 3.3V voltage supply supported |

**Bit 23 – SRSUP** Suspend/Resume Support

This bit indicates whether the peripheral supports the Suspend/Resume functionality. If this bit is set to 0, the user does not issue either Suspend or Resume commands because the Suspend and Resume mechanism (refer to “Suspend and Resume Mechanism” in the “SD Host Controller Simplified Specification V3.00” ) is not supported.

| Value | Description |
|---|---|
| 0 | Suspend/Resume not supported |
| 1 | Suspend/Resume supported |

**Bit 22 – SDMASUP** SDMA Support

This bit indicates whether the peripheral is capable of using SDMA to transfer data between system memory and the peripheral directly.

| Value | Description |
|---|---|
| 0 | SDMA not supported |
| 1 | SDMA supported |

**Bit 21 – HSSUP** High Speed Support

This bit indicates whether the peripheral and the system support high-speed mode and they can supply SDCLK frequency from 25 MHz to 50 MHz.

| Value | Description |
|---|---|
| 0 | High speed not supported |
| 1 | High speed supported |

**Bit 19 – ADMA2SUP** ADMA2 Support

This bit indicates whether the peripheral is capable of using ADMA2.

| Value | Description |
|---|---|
| 0 | ADMA2 not supported |
| 1 | ADMA2 supported |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 18 – ED8SUP** 8-Bit Support for Embedded Device
This bit indicates whether the peripheral is capable of using the 8-bit Bus Width mode.

| Value | Description |
|---|---|
| 0 | 8-bit bus width not supported |
| 1 | 8-bit bus width supported |

**Bits 17:16 – MAXBLKL[1:0]** Max Block Length
This field indicates the maximum block size that the user can read and write to the buffer in the peripheral.
**Note:** For SD Memory Cards, the transfer block length is always 512 bytes, regardless of this field.

| Value | Name | Description |
|---|---|---|
| 0 | 512 | 512 bytes |
| 1-3 | NONE | Reserved |

**Bits 15:8 – BASECLKF[7:0]** Base Clock Frequency
This value indicates the frequency of the base clock (BASECLK). The user uses this value to calculate the clock divider value (refer to SDCLK Frequency Select (SDCLKFSEL) in CCR).
If this field is set to 0, the user must get the information through another method. The GCLK Core clock (GCLK_SDHCx) divided by 2 is the Base Clock Frequency.
$F_{\mathrm{BASECLK}} = \mathrm{BASECLKF}_{\mathrm{MHz}}$

**Bit 7 – TEOCLKU** Timeout Clock Unit
This bit shows the unit of the base clock frequency used to detect Data Timeout Error.

| Value | Description |
|---|---|
| 0 | kHz |
| 1 | MHz |

**Bits 5:0 – TEOCLKF[5:0]** Timeout Clock Frequency
This bit shows the timeout clock frequency (TEOCLK) used to detect Data Timeout Error.
If this field is set to 0, the user must get the information through another method.
The Timeout Clock Unit (TEOCLKU) defines the unit of this field's value.
– TEOCLKU = 0:
$F_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{KHz}}$

– TEOCLKU = 1:
$F_{\mathrm{TEOCLK}} = \mathrm{TEOCLKF}_{\mathrm{MHz}}$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.27. Capabilities 1 Register

**Name:** CA1R
**Offset:** 0x44
**Reset:** 0x00000070
**Property:** -

**Note:** The Capabilities 1 Register is not supposed to be written by the user.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CLKMULT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | TSDR50 | | TCNTRT[3:0] | | | |
| Access | | | R/W | | R/W | R/W | R/W | R/W |
| Reset | | | 0 | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | DRVDSUP | DRVCSUP | DRVASUP | | DDR50SUP | SDR104SUP | SDR50SUP |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 23:16 – CLKMULT[7:0]** Clock Multiplier
This field indicates the multiplier factor between the Base Clock (BASECLK) used for the Divided Clock Mode and the Multiplied Clock (MULTCLK) used for the Programmable Clock mode (refer to CCR).
Reading this field to 0 means that the Programmable Clock mode is not supported.
$F_{\mathrm{MULTCLK}} = F_{\mathrm{BASECLK}} \times (\mathrm{CLKMULT}+1)$

**Bit 13 – TSDR50** Use Tuning for SDR50
If this bit is set to 1, the peripheral requires tuning to operate SDR50 (tuning is always required to operate SDR104).

| Value | Description |
|---|---|
| 0 | SDR50 does not require tuning. |
| 1 | SDR50 requires tuning. |

**Bits 11:8 – TCNTRT[3:0]** Timer Count For Re-Tuning
This field indicates an initial value of the Re-Tuning Timer for Re-Tuning Mode (RTMODE) 1 to 3. Reading this field at 0 means that the Re-Tuning Timer is disabled. The Re-Tuning Timer initial value ranges from 0 to 1024 seconds.
$T_{\mathrm{TIMER}} = 2^{(\mathrm{TCNTRT}+-1)}{}_{\mathrm{Seconds}}$

**Bit 6 – DRVDSUP** Driver Type D Support

| Value | Description |
|---|---|
| 0 | Driver type D is not supported. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 5 – DRVCSUP** Driver Type C Support

| Value | Description |
|---|---|
| 0 | Driver type C is not supported. |

**Bit 4 – DRVASUP** Driver Type A Support

| Value | Description |
|---|---|
| 0 | Driver type A is not supported. |

**Bit 2 – DDR50SUP** DDR50 Support

| Value | Description |
|---|---|
| 0 | DDR50 mode is not supported. |

**Bit 1 – SDR104SUP** SDR104 Support

| Value | Description |
|---|---|
| 0 | SDR104 mode is not supported. |
| 1 | SDR104 mode is supported. |

**Bit 0 – SDR50SUP** SDR50 Support

| Value | Description |
|---|---|
| 0 | SDR50 mode is not supported. |
| 1 | SDR50 mode is supported. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 40.8.28. Maximum Current Capabilities Register

**Name:** MCCAR
**Offset:** 0x48
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | MAXCUR18V[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MAXCUR30V[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | MAXCUR33V[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – MAXCUR18V[7:0]** Maximum Current for 1.8V
This field indicates the maximum current capability for 1.8V voltage. This value is meaningful only if V18VSUP is set to 1 in CA0RCA1R. Reading MAXCUR18V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR18}V$

**Bits 15:8 – MAXCUR30V[7:0]** Maximum Current for 3.0V
This field indicates the maximum current capability for 3.0V voltage. This value is meaningful only if V30VSUP is set to 1 in CA0R. Reading MAXCUR30V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR30}V$

**Bits 7:0 – MAXCUR33V[7:0]** Maximum Current for 3.3V
This field indicates the maximum current capability for 3.3V voltage. This value is meaningful only if V33VSUP is set to 1 in CA0R. Reading MAXCUR33V at 0 means that the user must get information via another method.
$\mathrm{Imax}_{\mathrm{mA}} = 4 \times \mathrm{MAXCURR33}V$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.29. Force Event Register for Auto CMD Error Status

**Name:** FERACES
**Offset:** 0x50
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMDNI | | | ACMDIDX | ACMDEND | ACMDCRC | ACMDTEO | ACMD12NE |
| Access | W | | | W | W | W | W | W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – CMDNI** Force Event for Command Not Issued by Auto CMD12 Error
For testing purposes, the user can write this bit to 1 to rise the CMDNI status flag in ACESR.
Writing this bit to 0 has no effect.

**Bit 4 – ACMDIDX** Force Event for Auto CMD Index Error
For testing purposes, the user can write this bit to 1 to rise the ACMDIDX status flag in ACESR.
Writing this bit to 0 has no effect.

**Bit 3 – ACMDEND** Force Event for Auto CMD End Bit Error
For testing purposes, the user can write this bit to 1 to rise the ACMDEND status flag in ACESR.
Writing this bit to 0 has no effect.

**Bit 2 – ACMDCRC** Force Event for Auto CMD CRC Error
For testing purposes, the user can write this bit to 1 to rise the ACMDCRC status flag in ACESR.
Writing this bit to 0 has no effect.

**Bit 1 – ACMDTEO** Force Event for Auto CMD Timeout Error
For testing purposes, the user can write this bit to 1 to rise the ACMDTEO status flag in ACESR.
Writing this bit to 0 has no effect.

**Bit 0 – ACMD12NE** Force Event for Auto CMD12 Not Executed
For testing purposes, the user can write this bit to 1 to rise the ACMD12NE status flag in ACESR.
Writing this bit to 0 has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.30. Force Event Register for Error Interrupt Status

**Name:** FEREIS
**Offset:** 0x52
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | BOOTAE | | | ADMA | ACMD |
| Access | | | | W | | | W | W |
| Reset | | | | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CURLIM | DATEND | DATCRC | DATTEO | CMDIDX | CMDEND | CMDCRC | CMDTEO |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 12 – BOOTAE** Force Event for Boot Acknowledge Error
For testing purposes, the user can write this bit to 1 to rise the BOOTAE status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 9 – ADMA** Force Event for ADMA Error
For testing purposes, the user can write this bit to 1 to rise the ADMA status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 8 – ACMD** Force Event for Auto CMD Error
For testing purposes, the user can write this bit to 1 to rise the ACMD status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 7 – CURLIM** Force Event for Current Limit Error
For testing purposes, the user can write this bit to 1 to rise the CURLIM status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 6 – DATEND** Force Event for Data End Bit Error
For testing purposes, the user can write this bit to 1 to rise the DATEND status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 5 – DATCRC** Force Event for Data CRC error
For testing purposes, the user can write this bit to 1 to rise the DATCRC status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 4 – DATTEO** Force Event for Data Timeout error
For testing purposes, the user can write this bit to 1 to rise the DATTEO status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 3 – CMDIDX** Force Event for Command Index Error
For testing purposes, the user can write this bit to 1 to rise the CMDIDX status flag in EISTR.
Writing this bit to 0 has no effect.

**Bit 2 – CMDEND** Force Event for Command End Bit Error
For testing purposes, the user can write this bit to 1 to rise the CDMEND status flag in EISTR.
Writing this bit to 0 has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 1 – CMDCRC** Force Event for Command CRC Error
For testing purposes, the user can write this bit to 1 to rise the CMDCRC status flag in EISTR. Writing this bit to 0 has no effect.

**Bit 0 – CMDTEO** Force Event for Command Timeout Error
For testing purposes, the user can write this bit to 1 to rise the CMDTEO status flag in EISTR. Writing this bit to 0 has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 40.8.31. ADMA Error Status Register

**Name:** AESR
**Offset:** 0x54
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | LMIS | ERRST[1:0] | |
| Access | | | | | | R | R | R |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – LMIS** ADMA Length Mismatch Error
This error occurs in the following two cases:

- While Block Count Enable (BCEN) is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count (BLKCNT) and Transfer Block Size (BLKSIZE).
- The total data length cannot be divided by the Transfer Block Size (BLKSIZE).

| Value | Description |
|---|---|
| 0 | No error |
| 1 | Error |

**Bits 1:0 – ERRST[1:0]** ADMA Error State
This field indicates the state of ADMA when an error has occurred during an ADMA data transfer. This field never indicates 2 because ADMA never stops in this state.

| Value | Name | Description |
|---|---|---|
| 0x0 | ST_STOP (Stop DMA) | Points to the descriptor following the error descriptor |
| 0x1 | ST_FDS (Fetch Descriptor) | Points to the error descriptor |
| 0x2 | - | Reserved |
| 0x3 | ST_TRF (Transfer Data) | Points to the descriptor following the error descriptor |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.32. ADMA System Address Register

**Name:** ASAR
**Offset:** 0x58
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADMASA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ADMASA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ADMASA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ADMASA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – ADMASA[31:0]** ADMA System Address

This field holds the byte address of the executing command of the descriptor table. At the start of ADMA, the user must set the start address of the descriptor table. The ADMA increments this register address, which points to the next Descriptor line to be fetched.
When the ADMA Error (ADMA) status flag rises, this field holds a valid descriptor address depending on the ADMA Error State (ERRST). The user must program Descriptor Table on 32-bit boundary and set 32-bit boundary address to this register. ADMA2 ignores the lower 2 bits of this register and assumes it to be 0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.33. Preset Value Register

**Name:** PVR
**Offset:** 0x60 + n*0x02 [n=0..7]
**Reset:** 0x0000
**Property:** Read/Write

One of the Preset Value Registers is effective based on the selected bus speed mode. The table below defines the conditions to select one of the PVRs.

**Table 40-3.** Preset Value Register Select Condition

| Selected Bus Speed Mode | VS18EN (HC2R) | HSEN (HC1R) | UHSMS (HC2R) |
|---|---|---|---|
| Default Speed | 0 | 0 | don't care |
| High Speed | 0 | Response Timeout Error | don't care |
| Reserved | 1 | don't care | Other values |

The following table shows the effective Preset Value Register according to the Selected Bus Speed mode.

**Table 40-4.** Preset Value Registers

| PVRx | Selected Bus Speed Mode | Signal Voltage |
|---|---|---|
| PVR0 | Initialization | 3.3V or 1.8V |
| PVR1 | Default Speed | 3.3V |
| PVR2 | High Speed | 3.3V |

When Preset Value Enable (PVALEN) in HC2R is set to 1, SDCLK Frequency Select (SDLCKFSEL) and Clock Generator Select (CLKGSEL) in CCR are automatically set based on the Selected Bus Speed mode. This means that the user does not need to set these fields when preset is enabled. A Preset Value Register for Initialization (PVR0) is not selected by Bus Speed mode. Before starting the initialization sequence, the user needs to set a clock preset value to SDCLKFSEL in CCR. PVALEN can be set to 1 after the initialization is completed.

**Note:** Preset Values in PVRx registers are not supposed to be written by the user. However, the user can modify preset values only if Capabilities Write Enable (CAPWREN) is written to 1 in CACR.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DRVSEL[1:0] | | | | | CLKGSEL | SDCLKFSEL[9:8] | |
| Access | R/W | R/W | | | | R/W | R/W | R/W |
| Reset | 0 | 0 | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SDCLKFSEL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:14 – DRVSEL[1:0]** Driver Strength Select Value for Initialization
Refer to DRVSEL in HC2R.

**Bit 10 – CLKGSEL** Clock Generator Select Value for Initialization
Refer to CGGSEL in CCR.

**Bits 9:0 – SDCLKFSEL[9:0]** SDCLK Frequency Select Value for Initialization
Refer to SDCLKFSEL in CCR.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.34. Slot Interrupt Status Register

**Name:** SISR
**Offset:** 0xFC
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | INTSSL[1:0] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – INTSSL[1:0]** Interrupt Signal for Each Slot
These status bits indicate the logical OR of Interrupt Signals and WakeUp Signal for each peripheral instance in the device. INTSSL[x] corresponds to instance SDHCx. There are 2 instances in this device.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.35. Host Controller Version Register

**Name:** HCVR
**Offset:** 0xFE
**Reset:** 0x1802
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | VVER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SVER[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

**Bits 15:8 – VVER[7:0]** Vendor Version Number
Reserved. Value subject to change. No functionality associated.

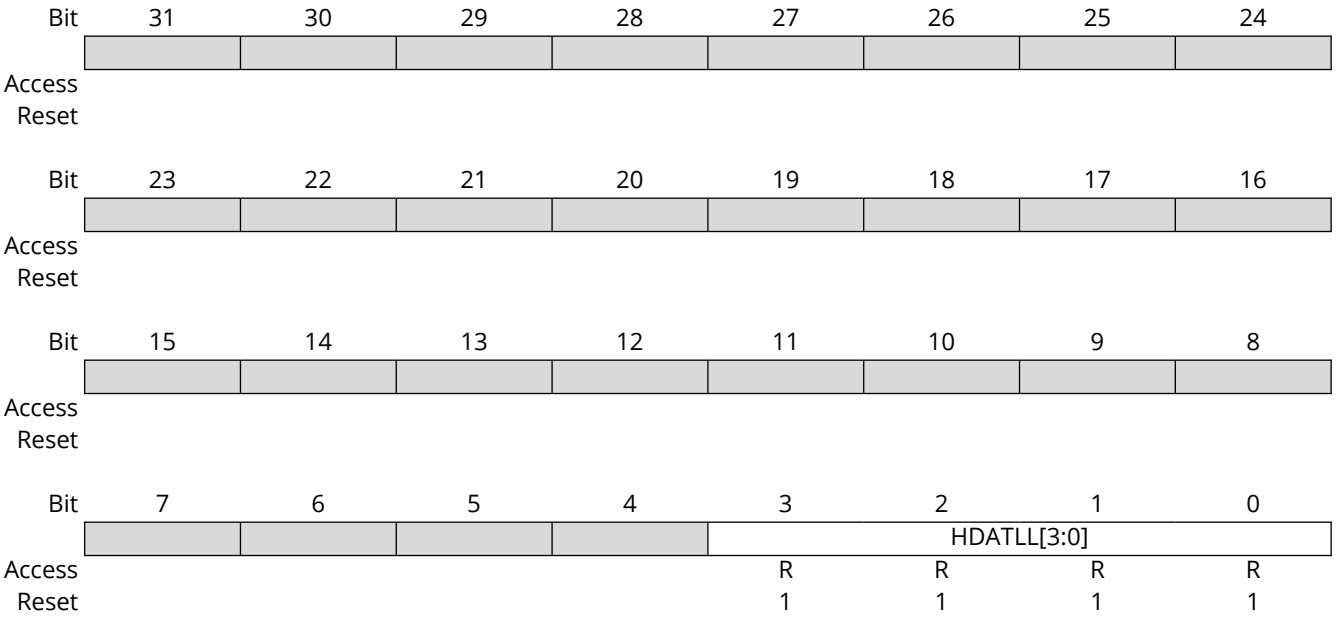
**Bits 7:0 – SVER[7:0]** Specification Version Number
This status indicates the SD Host Controller Specification Version.

| Value | Name |
|---|---|
| 0 | SD Host Specification Version 1.00 |
| 1 | SD Host Specification Version 2.00, including the feature of the ADMA and Test Register |
| 2 | SD Host Specification Version 3.00 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.36. Additional Present State Register

**Name:** APSR
**Offset:** 0x200
**Reset:** 0x0000000F
**Property:** -




**Bits 3:0 – HDATLL[3:0]** High Line Level
This status is used to check the DAT[7:4] line level to recover from errors, and for debugging.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.37. e.MMC Control 1 Register

**Name:** MC1R
**Offset:** 0x204
**Reset:** 0x00
**Property:** R/W

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FCD | RSTN | BOOTA | OPD | DDR | | CMDTYP[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |

**Bit 7 – FCD** e.MMC Force Card Detect
When using e.MMC, the user can set this bit to 1 to bypass the card detection procedure using the CD signal.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLED | e.MMC Forced Card Detect is disabled. The CD signal is used and debounce timing is applied. |
| 1 | ENABLED | e.MMC Forced Card Detect is enabled. |

**Bit 6 – RSTN** e.MMC Reset Signal
This bit controls the e.MMC reset signal.

| Value | Description |
|---|---|
| 0 | Reset signal is inactive. |
| 1 | Reset signal is active. |

**Bit 5 – BOOTA** e.MMC Boot Acknowledge Enable
This bit must be set according to the value of BOOT_ACK in the Extended CSD Register (refer to "Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51" ).
When this bit is set to 1, the peripheral waits for boot acknowledge pattern from the e.MMC before receiving boot data.
If the boot acknowledge pattern is wrong, the BOOTAE status flag rises in EISTR if BOOTAE is set in EISTER. An interrupt is generated if BOOTAE is set in EISIER.
If the no boot acknowledge pattern is received, the DATTEO status flag rises in EISTR if DATTEO is set in EISTER. An interrupt is generated if DATTEO is set in EISIER.

**Bit 4 – OPD** e.MMC Open Drain Mode
This bit sets the command line in open drain.

| Value | Description |
|---|---|
| 0 | The command line is in push-pull. |
| 1 | The command line is in open drain. |

**Bit 3 – DDR** e.MMC HSDDR Mode
This bit selects the High Speed DDR mode.

| Value | Description |
|---|---|
| 0 | High Speed DDR is not selected. |
| 1 | High Speed DDR is selected.<br>Note: The clock divider (DIV) in CCR must be set to a value different from 0 when HSEN is 1. |

**Bits 1:0 – CMDTYP[1:0]** e.MMC Command Type


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0 | NORMAL | The command is not an e.MMC specific command. |
| 1 | WAITIRQ | This bit must be set to 1 when the e.MMC is in Interrupt mode (CMD40). Refer to “Interrupt Mode” in the “Embedded MultiMedia Card (e.MMC) Electrical Standard 4.51” . |
| 2 | STREAM | This bit must be set to 1 in the case of Stream Read(CMD11) or Stream Write (CMD20). Only effective for e.MMC up to revision 4.41. |
| 3 | BOOT | Starts a Boot Operation mode at the next write to CR. Boot data are read directly from e.MMC device. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 40.8.38. e.MMC Control 2 Register

**Name:** MC2R
**Offset:** 0x205
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ABOOT | SRESP |
| Access | | | | | | | W | W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ABOOT** e.MMC Abort Boot
This bit is used to exit from Boot mode. Writing this bit to 1 exits the Boot Operation mode. Writing 0 is ignored.

**Bit 0 – SRESP** e.MMC Abort Wait IRQ
This bit is used to exit from the Interrupt mode. When this bit is written to 1, the peripheral sends the CMD40 response automatically. This brings the e.MMC from Interrupt mode to the standard Data Transfer mode. Writing this bit to 0 is ignored.
Note: This bit is only effective when CMD_TYP in MC1R is set to WAITIRQ.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.39. AHB Control Register

**Name:** ACR
**Offset:** 0x208
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BMAX[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – BMAX[1:0]** AHB Maximum Burst
This field selects the maximum burst size in case of DMA transfer.

| Value | Name | Description |
|---|---|---|
| 0 | INCR16 | The maximum burst size is INCR16. |
| 1 | INCR8 | The maximum burst size is INCR8. |
| 2 | INCR4 | The maximum burst size is INCR4. |
| 3 | SINGLE | Only SINGLE transfers are performed. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.40. Clock Control 2 Register

**Name:** CC2R
**Offset:** 0x20C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | FSDCLKD |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – FSDCLKD** Force SDCLK Disabled

The user can choose to maintain the SDCLK during 8 SDCLK cycles after the end bit of the last data block in case of a read transaction, or after the end bit of the CRC status in case of a write transaction.

| Value | Description |
|---|---|
| 0 | The SDCLK is forced and it cannot be stopped immediately after the transaction. |
| 1 | The SDCLK is not forced and it can be stopped immediately after the transaction. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.41. Capabilities Control Register

**Name:** CACR
**Offset:** 0x230
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | KEY[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CAPWREN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bits 15:8 – KEY[7:0]** Key

| Value | Name | Description |
|---|---|---|
| 0x46 | KEY | Writing any other value in this field aborts the write operation of the CAPWREN bit.<br>Always reads as 0. |

**Bit 0 – CAPWREN** Capabilities Write Enable
This bit can only be written if KEY correspond to 0x46.

| Value | Description |
|---|---|
| 0 | Capabilities registers (CA0R and CA1R) cannot be written. |
| 1 | Capabilities registers (CA0R and CA1R) can be written. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 40.8.42. Debug Register

**Name:** DBGR
**Offset:** 0x234
**Reset:** 0x00
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | NIDBG |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – NIDBG** Non-Intrusive Debug

| Value | Name | Description |
|---|---|---|
| 0 | IDBG | Reading the BDPR via debugger increments the dual port RAM read pointer. |
| 1 | NIDBG | Reading the BDPR via debugger does not increment the dual port RAM read pointer. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 41. Configurable Custom Logic (CCL)

## 41.1. Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, to events, or to other internal peripherals. This allows the user to eliminate logic gates for simple glue logic functions on the PCB.

Each LookUp Table (LUT) consists of three inputs, a truth table, an optional synchronizer/filter, and an optional edge detector. Each LUT can generate an output as a user programmable logic expression with three inputs. Inputs can be individually masked.

The output can be combinatorially generated from the inputs, and can be filtered to remove spikes. Optional sequential logic can be used. The inputs of the sequential module are individually controlled by two independent, adjacent LUT (LUT0/LUT1, LUT2/LUT3 etc.) outputs, enabling complex waveform generation.

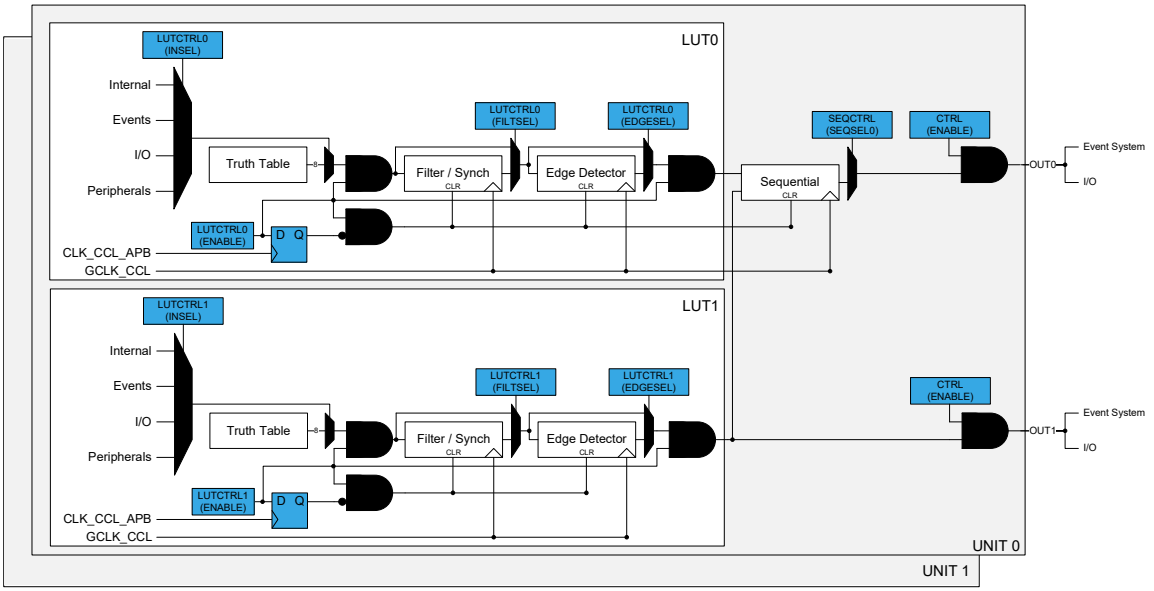
## 41.2. Features

- Glue logic for general purpose PCB design
- Up to 4 programmable LookUp Tables (LUTs)
- Combinatorial logic functions:
  AND, NAND, OR, NOR, XOR, XNOR, NOT
- Sequential logic functions:
  Gated D Flip-Flop, JK Flip-Flop, gated D Latch, RS Latch
- Flexible LUT inputs selection:
  - I/Os
  - Events
  - Internal peripherals
  - Subsequent LUT output
- Output can be connected to the I/O pins or the Event System
- Optional synchronizer, filter, or edge detector available on each LUT output


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 41.3. Block Diagram

**Figure 41-1.** Configurable Custom Logic




## 41.4. Signal Description

| Pin Name | Type | Description |
|---|---|---|
| OUT[3:0] | Digital output | Output from lookup table |
| IN[11:0] | Digital input | Input to lookup table |

For additional information on the pin mapping for this peripheral, refer to “I/O Multiplexing and Considerations”. One signal can be mapped on several pins.

**Related Links**

I/O Multiplexing and Considerations

## 41.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 41.5.1. I/O Lines

The CCL can take inputs and generate output through I/O pins. For this to function properly, the I/O pins must be configured to be used by a Look Up Table (LUT).

**Related Links**

PORT - I/O Pin Controller

### 41.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

**Related Links**

PM – Power Manager


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 41.5.3. Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the Main Clock module, MCLK (see *MCLK - Main Clock*), and the default state of CLK_CCL_APB can be found in *Peripheral Clock Masking*.

A generic clock (GCLK_CCL) is optionally required to clock the CCL. This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using input events, filter, edge detection or sequential logic. GCLK_CCL is required when input events, a filter, an edge detector, or a sequential sub-module is enabled. Refer to *GCLK - Generic Clock Controller* for details.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).

**Related Links**
MCLK – Main Clock
Peripheral Clock Masking
GCLK - Generic Clock Controller

### 41.5.4. DMA

Not applicable.

### 41.5.5. Interrupts

Not applicable.

### 41.5.6. Events

The CCL can use events from other peripherals and generate events that can be used by other peripherals. For this feature to function, the events have to be configured properly. Refer to the Related Links below for more information about the event users and event generators.

**Related Links**
Event System (EVSYS)

### 41.5.7. Debug Operation

When the CPU is halted in Debug mode the CCL continues normal operation. However, the CCL cannot be halted when the CPU is halted in Debug mode. If the CCL is configured in a way that requires it to be periodically serviced by the CPU, improper operation or data loss may result during debugging.

### 41.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC). Refer to *PAC - Peripheral Access Controller* for details.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 41.5.9. Analog Connections

Not applicable.

## 41.6. Functional Description

### 41.6.1. Principle of Operation

Configurable Custom Logic (CCL) is a programmable logic block that can use the device port pins, internal peripherals, and the internal Event System as both input and output channels. The CCL can


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

serve as glue logic between the device and external devices. The CCL can eliminate the need for external logic component and can also help the designer overcome challenging real-time constrains by combining core independent peripherals in clever ways to handle the most time critical parts of the application independent of the CPU.

## 41.6.2. Operation

### 41.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the corresponding even LUT is disabled (LUTCTRLx.ENABLE=0):

- Sequential Selection bits in the Sequential Control x (SEQCTRLx.SEQSEL) register

The following registers are enable-protected, meaning that they can only be written when the corresponding LUT is disabled (LUTCTRLx.ENABLE=0):

- LUT Control x (LUTCTRLx) register, except the ENABLE bit

Enable-protected bits in the LUTCTRLx registers can be written at the same time as LUTCTRLx.ENABLE is written to '1', but not at the same time as LUTCTRLx.ENABLE is written to '0'.

Enable-protection is denoted by the Enable-Protected property in the register description.

### 41.6.2.2. Enabling, Disabling, and Resetting

The CCL is enabled by writing a '1' to the Enable bit in the Control register (CTRL.ENABLE). The CCL is disabled by writing a '0' to CTRL.ENABLE.

Each LUT is enabled by writing a '1' to the Enable bit in the LUT Control x register (LUTCTRLx.ENABLE). Each LUT is disabled by writing a '0' to LUTCTRLx.ENABLE.

The CCL is reset by writing a '1' to the Software Reset bit in the Control register (CTRL.SWRST). All registers in the CCL will be reset to their initial state, and the CCL will be disabled. Refer to CTRL for details.

### 41.6.2.3. Lookup Table Logic

The lookup table in each LUT unit can generate any logic expression OUT as a function of three inputs (IN[2:0]), as shown in Figure 41-2. One or more inputs can be masked. The truth table for the expression is defined by TRUTH bits in LUT Control x register (LUTCTRLx.TRUTH).

**Figure 41-2.** Truth Table Output Value Selection

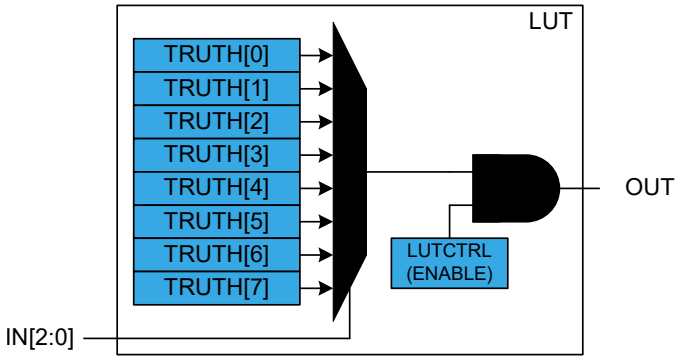



**Table 41-1.** Truth Table of LUT

| IN[2] | IN[1] | IN[0] | OUT |
|---|---|---|---|
| 0 | 0 | 0 | TRUTH[0] |
| 0 | 0 | 1 | TRUTH[1] |
| 0 | 1 | 0 | TRUTH[2] |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 41-1.** Truth Table of LUT (continued)

| IN[2] | IN[1] | IN[0] | OUT |
|---|---|---|---|
| 0 | 1 | 1 | TRUTH[3] |
| 1 | 0 | 0 | TRUTH[4] |
| 1 | 0 | 1 | TRUTH[5] |
| 1 | 1 | 0 | TRUTH[6] |
| 1 | 1 | 1 | TRUTH[7] |

### 41.6.2.4. Truth Table Inputs Selection

#### Input Overview

The inputs can be individually:

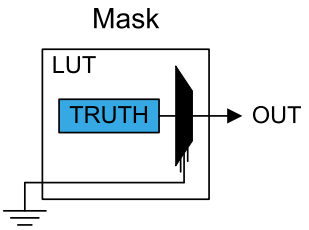
- Masked
- Driven by peripherals:
  - Analog comparator output (AC)
  - Timer/Counters waveform outputs (TC)
  - Serial Communication output transmit interface (SERCOM)
- Driven by internal events from Event System
- Driven by other CCL sub-modules

The Input Selection for each input y of LUT x is configured by writing the Input y Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

#### Masked Inputs (MASK)

When a LUT input is masked (LUTCTRLx.INSELy=MASK), the corresponding TRUTH input (IN) is internally tied to zero, as shown in this figure:

**Figure 41-3.** Masked Input Selection




#### Internal Feedback Inputs (FEEDBACK)

When selected (LUTCTRLx.INSELy=FEEDBACK), the Sequential (SEQ) output is used as input for the corresponding LUT.

The output from an internal sequential sub-module can be used as input source for the LUT, see figure below for an example for LUT0 and LUT1. The sequential selection for each LUT follows the formula:

$\mathrm{IN}[2\mathrm{N}][i] = \mathrm{SEQ}[N]$

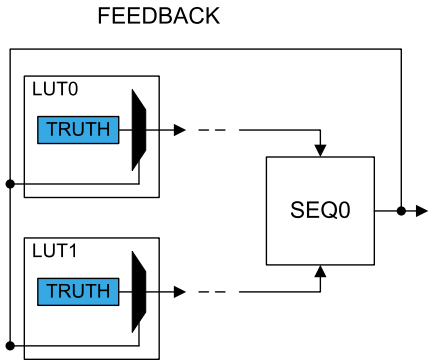
$\mathrm{IN}[2\mathrm{N}+1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i*=0,1,2 representing the LUT input index.

For details, refer to Sequential Logic.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

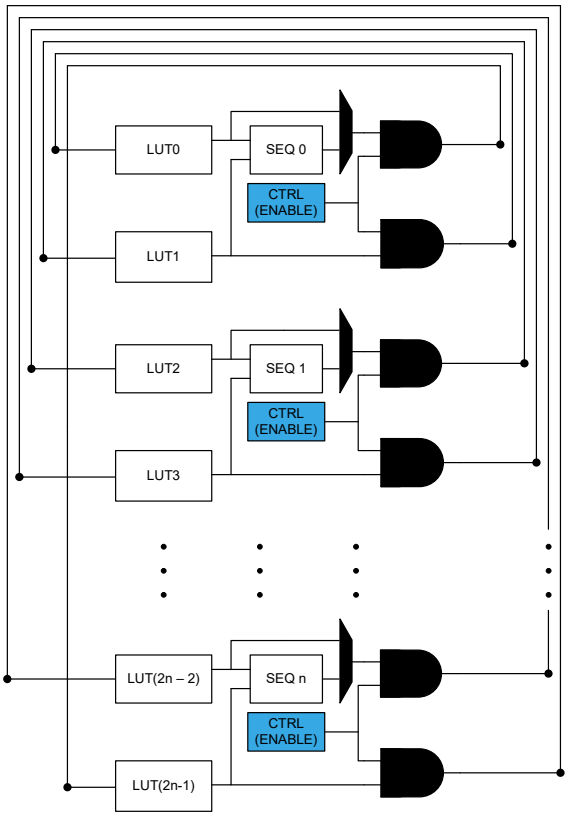
**Figure 41-4.** Feedback Input Selection




### Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (e.g., LUT2 is the input for LUT1), as shown in this figure:

**Figure 41-5.** Linked LUT Input Selection




### Internal Events Inputs Selection (EVENT)

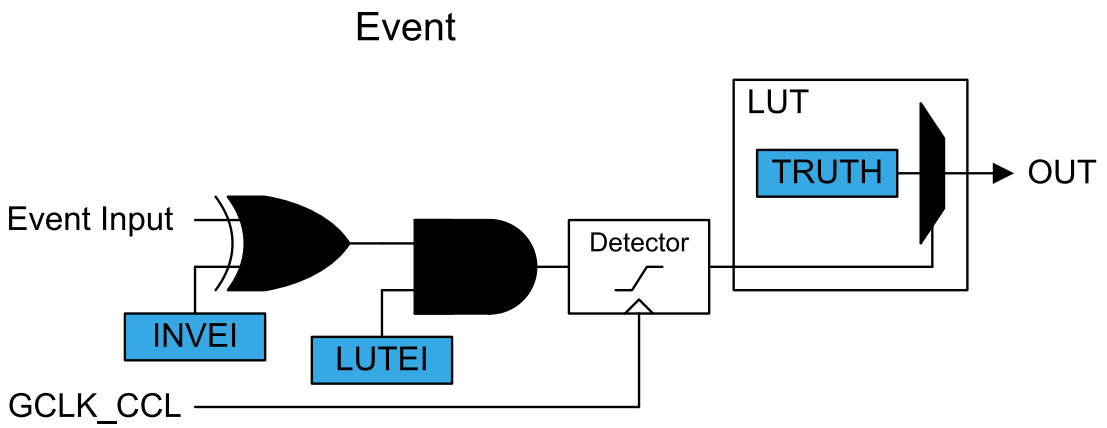
Asynchronous events from the Event System can be used as input selection, as shown in the below image. For each LUT, one event input line is available and can be selected on each LUT input. Before enabling the event selection by writing LUTCTRLx.INSELy=EVENT, the Event System must be configured first.

By default CCL includes an edge detector. When the event is received, an internal strobe is generated when a rising edge is detected. The pulse duration is one GCLK_CCL clock cycle. The following steps ensure proper operation:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

1. Enable the GCLK_CCL clock.
2. Configure the Event System to route the event asynchronously.
3. Select the event input type (LUTCTRLx.INSEL).
4. If a strobe must be generated on the event input falling edge, write a '1' to the Inverted Event Input Enable bit in LUT Control register (LUTCTRLx.INVEI) .
5. Enable the event input by writing the Event Input Enable bit in LUT Control register (LUTCTRLx.LUTEI) to '1'.

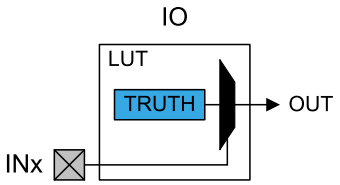
**Figure 41-6.** Event Input Selection




### I/O Pin Inputs (IO)

When the IO pin is selected as LUT input (LUTCTRLx.INSELy=IO), the corresponding LUT input will be connected to the pin, as shown in the figure below.

**Figure 41-7.** I/O Pin Input Selection




### Analog Comparator Inputs (AC)

The AC outputs can be used as input source for the LUT (LUTCTRLx.INSELy=AC).

The analog comparator outputs are distributed following the formula:

```
IN[N][i]=AC[N % ComparatorOutput_Number]
```

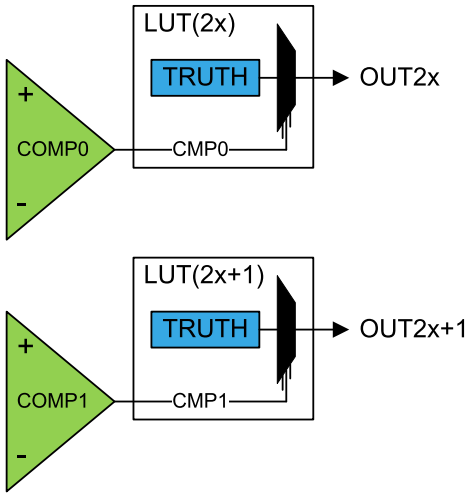
With *N* representing the LUT number and *i*=[0,1,2] representing the LUT input index.

Before selecting the comparator output, the AC must be configured first.

The output of comparator 0 is available on even LUTs ("LUT(2x)": LUT0, LUT2) and the comparator 1 output is available on odd LUTs ("LUT(2x+1)": LUT1, LUT3), as shown in the figure below.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-8.** AC Input Selection




### Timer/Counter Inputs (TC)

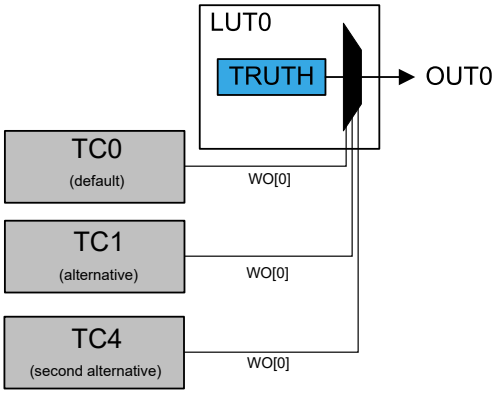
The TC waveform output WO[0] can be used as input source for the LUT (LUTCTRLx.INSELy=TC). Only consecutive instances of the TC, i.e. TCx and the subsequent TC(x+1), are available as default and alternative TC selections (e.g., TC0 and TC1 are sources for LUT0, TC1 and TC2 are sources for LUT1, etc). See the figure below for an example for LUT0. More general, the Timer/Counter selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = DefaultTC[N \,\%\, \mathrm{TC_Instance_Number}]$

$\mathrm{IN}[N][i] = AlternativeTC[(N+1) \,\%\, \mathrm{TC_Instance_Number}]$

Where N represents the LUT number and i represents the LUT input index (i=0,1,2).

**Figure 41-9.** TC Input Selection




### Timer/Counter for Control Application Inputs (TCC)

The TCC waveform outputs can be used as input source for the LUT. Only WO[2:0] outputs can be selected and routed to the respective LUT input (i.e., IN0 is connected to WO0, IN1 to WO1, and IN2 to WO2), as shown in the figure below.

**Note:**
The TCC selection for each LUT follows the formula:

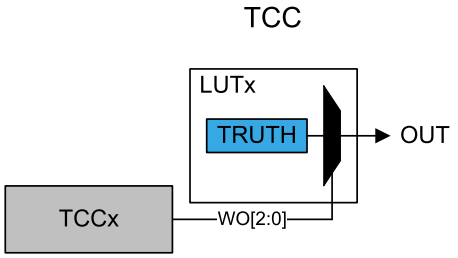
$\mathrm{IN}[N][i] = TCC[N \,\%\, TCC_\mathrm{Instance_Number}]$

Where *N* represents the LUT number.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Before selecting the waveform outputs, the TCC must be configured first.

**Figure 41-10.** TCC Input Selection




### Serial Communication Output Transmit Inputs (SERCOM)

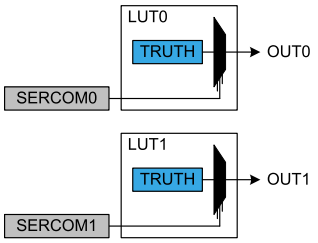
The serial engine transmitter output from Serial Communication Interface (SERCOM TX, TXd for USART, MOSI for SPI) can be used as input source for the LUT. The figure below shows an example for LUT0 and LUT1. The SERCOM selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = SERCOM[N \,\%\, \mathrm{SERCOM_Instance_Number}]$

With *N* representing the LUT number and *i*=0,1,2 representing the LUT input index.

Before selecting the SERCOM as input source, the SERCOM must be configured first: the SERCOM TX signal must be output on SERCOMn/pad[0], which serves as input pad to the CCL.

**Figure 41-11.** SERCOM Input Selection




### Related Links

I/O Multiplexing and Considerations
PORT - I/O Pin Controller
GCLK - Generic Clock Controller
AC – Analog Comparators
Timer/Counter (TC)
TCC – Timer/Counter for Control Applications
SERCOM – Serial Communication Interface

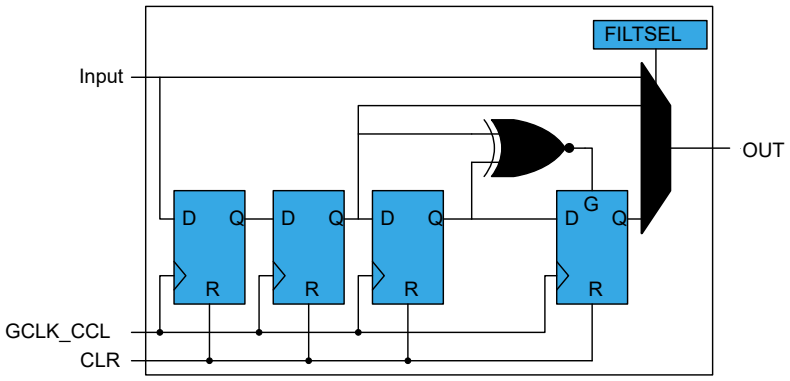
## 41.6.2.5. Filter

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed by clocking through filters, if demanded by application needs.

The Filter Selection bits in LUT Control register (LUTCTRLx.FILTSEL) define the synchronizer or digital filter options. When a filter is enabled, the OUT output will be delayed by two to five GCLK cycles. One APB clock after the corresponding LUT is disabled, all internal filter logic is cleared.
**Note:** Events used as LUT input will also be filtered, if the filter is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-12.** Filter



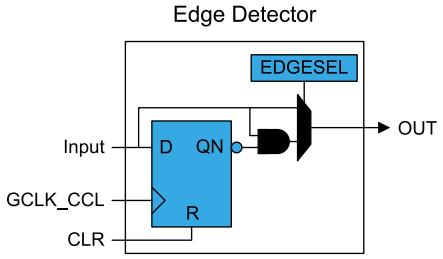

### 41.6.2.6. Edge Detector

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the TRUTH table should be inverted.

The edge detector is enabled by writing '1' to the Edge Selection bit in LUT Control register (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, either the filter or synchronizer must be enabled.

Edge detection is disabled by writing a '0' to LUTCTRLx.EDGESEL. After disabling a LUT, the corresponding internal Edge Detector logic is cleared one APB clock cycle later.

**Figure 41-13.** Edge Detector




### 41.6.2.7. Sequential Logic

Each LUT pair can be connected to the internal sequential logic which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL clock and optionally each LUT filter or edge detector must be enabled.

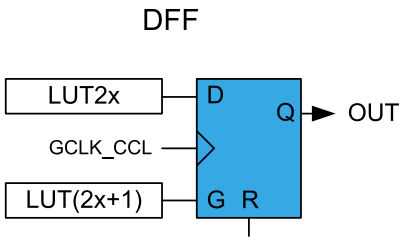
**Note:** While configuring the sequential logic, the even LUT must be disabled. When configured the even LUT must be enabled.

#### Gated D Flip-Flop (DFF)

When the DFF is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 41-14.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-14.** D Flip Flop




When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 41-2.

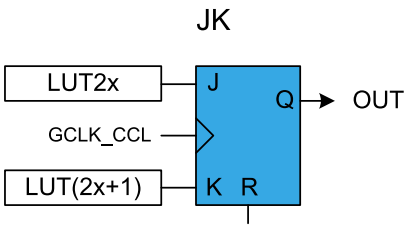
**Table 41-2.** DFF Characteristics

| R | G | D | OUT |
|---|---|---|---|
| 1 | X | X | Clear |
| 0 | 1 | 1 | Set |
| | | 0 | Clear |
| | 0 | X | Hold state (no change) |

### JK Flip-Flop (JK)

When this configuration is selected, the J-input is driven by the even LUT output (LUT0 and LUT2), and the K-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 41-15.

**Figure 41-15.** JK Flip Flop




When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the flip-flop is asynchronously cleared. The reset command (R) is kept enabled for one APB clock cycle. In all other cases, the flip-flop output (OUT) is refreshed on rising edge of the GCLK_CCL, as shown in Table 41-3.

**Table 41-3.** JK Characteristics

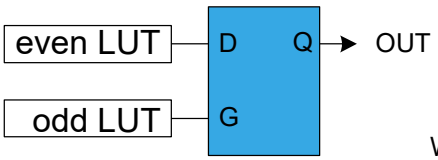
| R | J | K | OUT |
|---|---|---|---|
| 1 | X | X | Clear |
| 0 | 0 | 0 | Hold state (no change) |
| 0 | 0 | 1 | Clear |
| 0 | 1 | 0 | Set |
| 0 | 1 | 1 | Toggle |

### Gated D-Latch (DLATCH)

When the DLATCH is selected, the D-input is driven by the even LUT output (LUT0 and LUT2), and the G-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 41-14.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-16.** D-Latch




When the even LUT is disabled (LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The G-input is forced enabled for one more APB clock cycle, and the D-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 41-4.

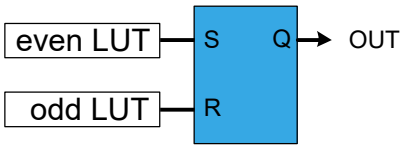
**Table 41-4.** D-Latch Characteristics

| G | D | OUT |
|---|---|---|
| 0 | X | Hold state (no change) |
| 1 | 0 | Clear |
| 1 | 1 | Set |

### RS Latch (RS)

When this configuration is selected, the S-input is driven by the even LUT output (LUT0 and LUT2), and the R-input is driven by the odd LUT output (LUT1 and LUT3), as shown in Figure 41-17.

**Figure 41-17.** RS-Latch




When the even LUT is disabled LUTCTRL0.ENABLE=0 / LUTCTRL2.ENABLE=0), the latch output will be cleared. The R-input is forced enabled for one more APB clock cycle and S-input to zero. In all other cases, the latch output (OUT) is refreshed as shown in Table 41-5.

**Table 41-5.** RS-Latch Characteristics

| S | R | OUT |
|---|---|---|
| 0 | 0 | Hold state (no change) |
| 0 | 1 | Clear |
| 1 | 0 | Set |
| 1 | 1 | Forbidden state |

## 41.6.3. Events

The CCL can generate the following output events:

- OUTx: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRL.LUTEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

The CCL can take the following actions on an input event:

- INSELx: The event is used as input for the TRUTH table. For further details refer to Events.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRL.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

### Related Links

Event System (EVSYS)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 41.6.4. Sleep Mode Operation

When using the GCLK_CCL internal clocking, writing the Run In Standby bit in the Control register (CTRL.RUNSTDBY) to '1' will allow GCLK_CCL to be enabled in Standby Sleep mode.

If CTRL.RUNSTDBY=0, the GCLK_CCL will be disabled in Standby Sleep mode. If the Filter, Edge Detector or Sequential logic are enabled, the LUT output will be forced to zero in STANDBY mode. In all other cases, the TRUTH table decoder will continue operation and the LUT output will be refreshed accordingly.

**Related Links**

PM – Power Manager


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

## 41.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRL | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| 0x01 ... 0x03 | Reserved | | | | | | | | | |
| 0x04 | SEQCTRL0 | 7:0 | | | | | SEQSEL[3:0] | | | |
| 0x05 | SEQCTRL1 | 7:0 | | | | | SEQSEL[3:0] | | | |
| 0x06 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | LUTCTRL0 | 7:0 | EDGESEL | | FILTSEL[1:0] | | | | ENABLE | |
| | | 15:8 | INSEL1[3:0] | | | | INSEL0[3:0] | | | |
| | | 23:16 | | LUTEO | LUTEI | INVEI | INSEL2[3:0] | | | |
| | | 31:24 | TRUTH[7:0] | | | | | | | |
| 0x0C | LUTCTRL1 | 7:0 | EDGESEL | | FILTSEL[1:0] | | | | ENABLE | |
| | | 15:8 | INSEL1[3:0] | | | | INSEL0[3:0] | | | |
| | | 23:16 | | LUTEO | LUTEI | INVEI | INSEL2[3:0] | | | |
| | | 31:24 | TRUTH[7:0] | | | | | | | |
| 0x10 | LUTCTRL2 | 7:0 | EDGESEL | | FILTSEL[1:0] | | | | ENABLE | |
| | | 15:8 | INSEL1[3:0] | | | | INSEL0[3:0] | | | |
| | | 23:16 | | LUTEO | LUTEI | INVEI | INSEL2[3:0] | | | |
| | | 31:24 | TRUTH[7:0] | | | | | | | |
| 0x14 | LUTCTRL3 | 7:0 | EDGESEL | | FILTSEL[1:0] | | | | ENABLE | |
| | | 15:8 | INSEL1[3:0] | | | | INSEL0[3:0] | | | |
| | | 23:16 | | LUTEO | LUTEI | INVEI | INSEL2[3:0] | | | |
| | | 31:24 | TRUTH[7:0] | | | | | | | |

## 41.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 41.8.1. Control

**Name:** CTRL
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | RUNSTDBY | | | | | ENABLE | SWRST |
| Access | | R/W | | | | | R/W | W |
| Reset | | 0 | | | | | 0 | 0 |

**Bit 6 – RUNSTDBY** Run in Standby
This bit indicates if the GCLK_CCL clock must be kept running in standby mode. The setting is ignored for configurations where the generic clock is not required. For details refer to Sleep Mode Operation.

**Important:** This bit must be written before enabling the CCL.

| Value | Description |
|---|---|
| 0 | Generic clock is not required in standby sleep mode. |
| 1 | Generic clock is required in standby sleep mode. |

**Bit 1 – ENABLE** Enable

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the CCL to their initial state.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 41.8.2. Sequential Control x

**Name:** SEQCTRL
**Offset:** 0x04 + n*0x01 [n=0..1]
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

**Note:** SEQCTRLx register is Enable-protected when LUTCTRLx.ENABLE = 1.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SEQSEL[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 3:0 – SEQSEL[3:0]** Sequential Selection
These bits select the sequential configuration:
Sequential Selection

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Sequential logic is disabled |
| 0x1 | DFF | D flip flop |
| 0x2 | JK | JK flip flop |
| 0x3 | LATCH | D latch |
| 0x4 | RS | RS latch |
| 0x5 - 0xF | | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 41.8.3. LUT Control n

**Name:** LUTCTRL
**Offset:** 0x08 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-protected

**Note:** The LUTCTRLn register is Enable Protected when CCL.LUTCTRLn.ENABLE = 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TRUTH[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | LUTEO | LUTEI | INVEI | INSEL2[3:0] | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | INSEL1[3:0] | | | | INSEL0[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EDGESEL | | FILTSEL[1:0] | | | | ENABLE | |
| Access | R/W | | R/W | R/W | | | R/W | |
| Reset | 0 | | 0 | 0 | | | 0 | |

**Bits 31:24 – TRUTH[7:0]** Truth Table
These bits define the value of truth logic as a function of inputs IN[2:0].

**Bit 22 – LUTEO** LUT Event Output Enable

| Value | Description |
|---|---|
| 0 | LUT event output is disabled. |
| 1 | LUT event output is enabled. |

**Bit 21 – LUTEI** LUT Event Input Enable

| Value | Description |
|---|---|
| 0 | LUT incoming event is disabled. |
| 1 | LUT incoming event is enabled. |

**Bit 20 – INVEI** Inverted Event Input Enable

| Value | Description |
|---|---|
| 0 | Incoming event is not inverted. |
| 1 | Incoming event is inverted. |

**Bits 8:11, 12:15, 16:19 – INSELx** LUT Input x Source Selection [x=0..2]
These bits select the LUT input x source:

| Value | Name | Description |
|---|---|---|
| 0x0 | MASK | Masked input |
| 0x1 | FEEDBACK | Feedback input source |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x2 | LINK | Linked LUT input source |
| 0x3 | EVENT | Event input source |
| 0x4 | IO | I/O pin input source |
| 0x5 | AC | AC input source: CMP[0] (LUT0) / CMP[1] (LUT1)/ CMP[2] (LUT2) / CMP[3] (LUT3) |
| 0x6 | TC | TC input source: TC0 (LUT0) / TC1 (LUT1)/ TC2 (LUT2) / TC3 (LUT3) |
| 0x7 | ALTTC | Alternative TC input source: TC1 (LUT0) / TC2 (LUT1) / TC3 (LUT2) / TC4 (LUT3) |
| 0x8 | TCC | TCC input source: TCC0 (LUT0) / TCC1 (LUT1) / TCC2 (LUT2) / TCC3 (LUT3) |
| 0x9 | SERCOM | SERCOM input source: SERCOM0 (LUT0) / SERCOM1 (LUT1)/ SERCOM2 (LUT2) / SERCOM3 (LUT3) |
| 0xA – 0xF | Reserved | Reserved |

**Bit 7 – EDGESEL** Edge Selection

| Value | Description |
|---|---|
| 0 | Edge detector is disabled. |
| 1 | Edge detector is enabled. |

**Bits 5:4 – FILTSEL[1:0]** Filter Selection

These bits select the LUT output filter options:
Filter Selection

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Filter disabled |
| 0x1 | SYNCH | Synchronizer enabled |
| 0x2 | FILTER | Filter enabled |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE** LUT Enable

| Value | Description |
|---|---|
| 0 | The LUT is disabled. |
| 1 | The LUT is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 42. AES – Advanced Encryption Standard

## 42.1. Overview

The Advanced Encryption Standard (AES) peripheral provides a means for symmetric-key encryption of 128-bit blocks, in compliance to NIST specifications.

The symmetric-key algorithm requires the same key for both encryption and decryption. Different key sizes are supported. The key size determines the number of repetitions of transformation rounds that convert the input (plaintext) into the final output (ciphertext). The number of repetition rounds is as follows:

- 10 rounds of repetition for 128-bit keys
- 12 rounds of repetition for 192-bit keys
- 14 rounds of repetition for 256-bit keys

## 42.2. Features

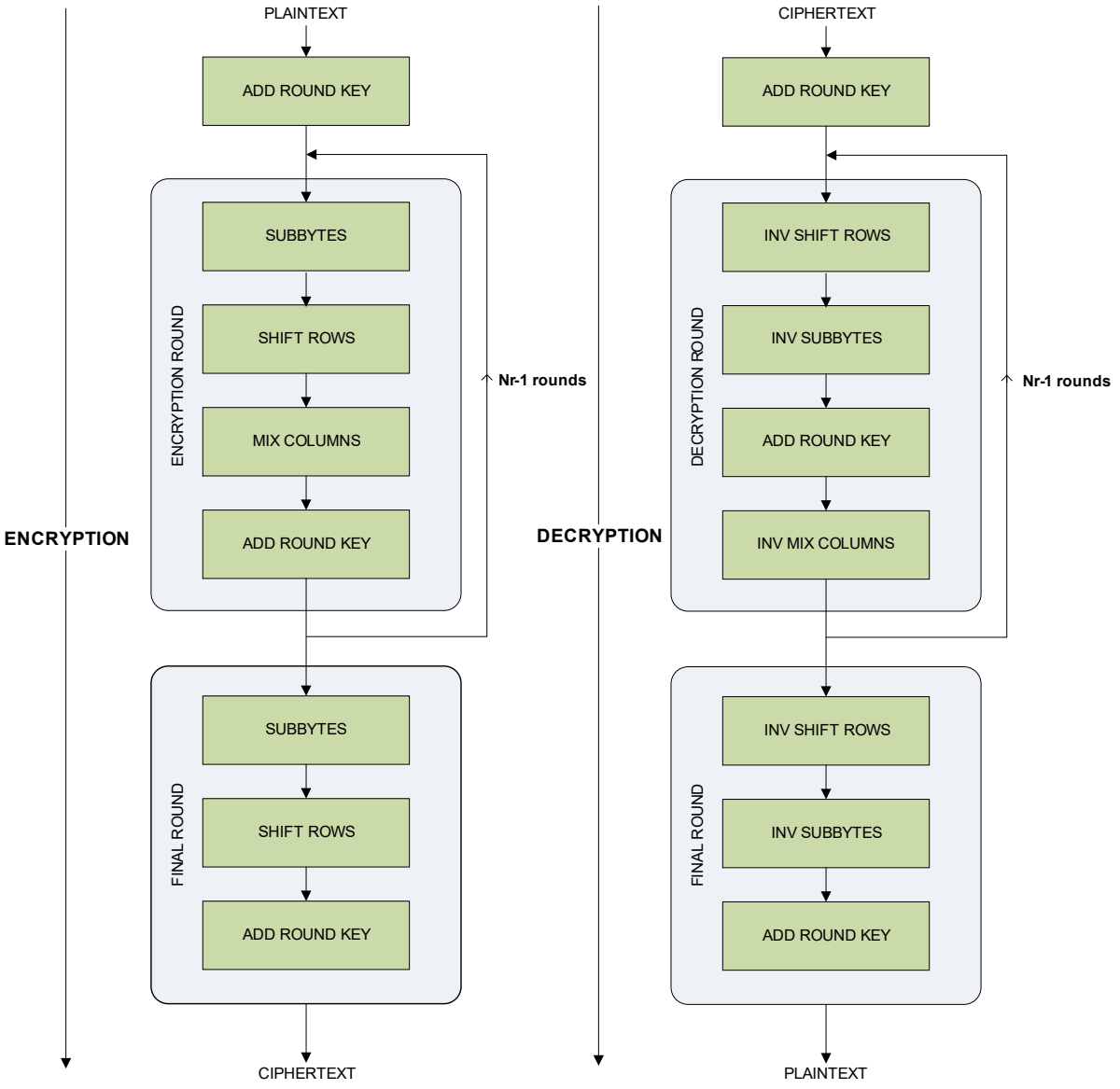
The following are key features of the AES module:

- Compliant with FIPS Publication 197, Advanced Encryption Standard (AES)
- 128/192/256 bit cryptographic key supported
- Encryption time of 57/67/77 cycles with 128-bit/192-bit/256-bit cryptographic key
- Five confidentiality modes of operation as recommended in NIST Special Publication 800-38A
- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)
- Supports Counter with CBC-MAC (CCM/CCM*) mode for authenticated encryption
- 8, 16, 32, 64, 128-bit data sizes possible in CFB mode
- Galois Counter Mode (GCM) encryption and authentication


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 42.3. Block Diagram

**Figure 42-1.** AES Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 42.4. Signal Description

Not applicable.

## 42.5. Product Dependencies

In order to use this AES module, other parts of the system must be configured correctly, as described below.

### 42.5.1. I/O Lines

Not applicable.

### 42.5.2. Power Management

The AES will continue to operate in Standby sleep mode, if it's source clock is running.

The AES interrupts can be used to wake up the device from Standby sleep mode. Refer to the Power Manager chapter for details on the different sleep modes.

AES is clocked only on the following conditions:

- When the DMA is enabled.
- Whenever there is an APB access for any read and write operation to the AES registers. (Not in Standby sleep mode.)
- When the AES is enabled & encryption/decryption is ongoing.

### 42.5.3. Clocks

The AES bus clock (CLK_AES_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_AES_APB can be found in *Peripheral Clock Masking*. The module is fully clocked by CLK_AES_APB.

**Related Links**

Peripheral Clock Masking

### 42.5.4. DMA

The AES has two DMA request lines: one for input data and one for output data. They are connected to the DMA Controller (DMAC). These DMA request triggers will be acknowledged by the DMAC ACK signals. Using the AES DMA requests requires the DMA Controller to be configured first. Refer to the device DMA documentation for additional information.

**Related Links:**

DMAC- Direct Memory Access Controller

### 42.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the AES interrupt requires the interrupt controller to be configured first. Refer to the Processor and Architecture chapter for details.

All the AES interrupts are synchronous wake-up sources. See *Sleep Mode Controller* for details.

**Related Links**

Sleep Mode Controller

### 42.5.6. Events

Not applicable.

### 42.5.7. Debug Operation

When the CPU is halted in Debug mode, the AES module continues normal operation. If the AES module is configured in a way that requires it to be periodically serviced by the CPU through


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

interrupts or similar, improper operation or data loss may result during debugging. The AES module can be forced to halt operation during debugging.

### 42.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the peripheral access controller (PAC), except the Interrupt Flag Register (INTFLAG) register.

Write-protection is denoted by the Write-Protected property in the register description.

Write-protection does not apply to accesses through an external debugger. Refer to *PAC - Peripheral Access Controller* chapter for additional information.

**Related Links**

PAC - Peripheral Access Controller

### 42.5.9. Analog Connections

Not applicable.

## 42.6. Functional Description

### 42.6.1. Principle of Operation

The following is an overview of the algorithm. Follow these steps:

- KeyExpansion: Round keys are derived from the CipherKey using Rijndael's key schedule.
- InitialRound:
  - AddRoundKey: Each byte of the state is combined with the round key using bitwise XOR.
- Rounds:
  - SubBytes: A non-linear substitution step where each byte is replaced with another according to a lookup table.
  - ShiftRows: A transposition step where each row of the state is shifted cyclically a certain number of steps.
  - MixColumns: A mixing operation which operates on the columns of the state, combining the four bytes in each column.
  - AddRoundKey
- Final Round (no MixColumns):
  - SubBytes
  - ShiftRows
  - AddRoundKey

The relationship between the module's clock frequency and throughput (in bytes per second) is given by:

Clock Frequency = (Throughput/2) x (Nr+1) for 2 byte parallel processing

Clock Frequency = (Throughput/4) x (Nr+1) for 4 byte parallel processing

Where, Nr represents the number of rounds, depending on the key length.

### 42.6.2. Basic Operation

#### 42.6.2.1. Initialization

The Control A (CTRLA) register is enable-protected.

Enable-protection is denoted by the Enable-Protected property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 42.6.2.2. Enabling, Disabling, and Resetting

The AES module is enabled by writing a one to the Enable bit in the Control A register (CTRLA.ENABLE). The module is disabled by writing a zero to CTRLA.ENABLE. The module is reset by writing a one to the Software Reset bit in the Control A register (CTRLA.SWRST).

#### 42.6.2.3. Basic Programming

The CIPHER bit in the Control A Register (CTRLA.CIPHER) allows selection between the encryption and the decryption processes. The AES is capable of using cryptographic keys of 128/192/256 bits to encrypt and decrypt data in blocks of 128 bits. The Key Size (128/192/256) can be programmed in the KEYSIZE field in the Control A Register (CTRLA.KEYSIZE). This 128-bit/192-bit/256-bit key is defined in the Key Word Registers (KEYWORD). By setting the XORKEY bit of CTRLA register, keyword can be updated with the resulting XOR value of user keyword and previous keyword content.

The input data for processing is written to a data buffer consisting of four 32-bit registers through the Data register address. The data buffer register (note that input and output data shares the same data buffer register) that is written to when the next write is performed is indicated by the Data Pointer in the Data Buffer Pointer (DATABUFPTR) register. This field is incremented by one or wrapped by hardware when a write to the INDATA register address is performed. This field can also be programmed, allowing the user direct control over which input buffer register to write. Note that when AES module is in the CFB operation mode with the data segment size less than 128 bits, the input data must be written to the first (DATABUFPTR = 0) and second (DATABUFPTR = 1) input buffer registers (see Table 42-1).

The input to the encryption processes of the CBC, CFB and OFB modes includes, in addition to the plaintext, a 128-bit data block called the Initialization Vector (IV), which must be set in the Initialization Vector Registers (INTVECT). Additionally, the GCM mode 128-bit authentication data needs to be programmed. The Initialization Vector is used in the initial step in the encryption of a message and in the corresponding decryption of the message. The Initialization Vector Registers are also used by the Counter mode to set the counter value.

It is necessary to notify AES module whenever the next data block it is going to process is the beginning of a new message. This is done by writing a one to the New Message bit in the Control B register (CTRLB.NEWMSG).

The AES modes of operation are selected by setting the AESMODE field in the Control A Register (CTRLA.AESMODE). In Cipher Feedback Mode (CFB), five data sizes are possible (8, 16, 32, 64 or 128 bits), configurable by means of the CFBS field in the Control A Register (CTRLA.CFBS). In Counter mode, the size of the block counter embedded in the module is 16 bits. Therefore, there is a rollover after processing 1 megabyte of data. The data pre-processing, post-processing and data chaining for the concerned modes are automatically performed by the module.

When data processing has completed, the Encryption Complete bit in the Interrupt Flag register (INTFLAG.ENCCMP) is set by hardware (which triggers an interrupt request if the corresponding interrupt is enabled). The processed output data is read out through the Output Data register (INDATA) address from the data buffer consisting of four 32-bit registers. The data buffer register that is read when the next read is performed is indicated by the Data Pointer field in the Data Buffer Pointer register (DATABUFPTR). This field is incremented by one or wrapped by hardware when a read from the INDATA register address is performed. This field can be programmed, giving the user direct control over which output buffer register to read from. Note that when AES module is in the CFB operation mode with the data segment size less than 128 bits, the output data must be read from the first (DATABUFPTR = 0) and second (DATABUFPTR = 1) output buffer registers (see Table 42-1). The Encryption Complete bit (INTFLAG.ENCCMP) is cleared by hardware after the processed data has been read from the relevant output buffer registers.

**Table 42-1.** Relevant Input/Output Data Registers for Different Confidentiality Modes

| Confidentiality Mode | Relevant Input / Output Data Registers |
|---|---|
| ECB | All |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 42-1.** Relevant Input/Output Data Registers for Different Confidentiality Modes (continued)

| Confidentiality Mode | Relevant Input / Output Data Registers |
|---|---|
| CBC | All |
| OFB | All |
| 128-bit CFB | All |
| 64-bit CFB | First and Second |
| 32-bit CFB | First |
| 16-bit CFB | First |
| 8-bit CFB | First |
| CTR | All |

#### 42.6.2.4. Start Modes

The Start mode field in the Control A register (CTRLA.STARTMODE) allows the selection of encryption Start mode.

1. Manual Start mode
   . The sequence for Manual Start mode is as follows:
   a. Write the 128/192/256 bit key in the Key Register (KEYWORD).
   b. Write the initialization vector or counter in the Initialization Vector Register (INTVECT). The initialization vector concerns all modes except ECB.
   c. Enable interrupts in the Interrupt Enable Set register (INTENSET), depending on whether an interrupt is required or not at the end of processing.
   d. Write the data to be encrypted or decrypted in the Data Registers (INDATA).
   e. Set the START bit in the Control B register (CTRLB.START) to begin the encryption or the decryption process.
   f. When the processing completes, the Encryption Complete bit in the Interrupt Flag register (INTFLAG.ENCCMP) raises. If Encryption Complete Interrupt has been enabled, the interrupt line of the AES is activated.
   g. When the software reads one of the Output Data registers (INDATA), the INTFLAG.ENCCMP bit is automatically cleared.
2. Auto Start mode.
   This mode is similar to the manual one, but as soon as the correct number of input data registers is written, processing is automatically started without setting the START bit in the Control B Register. The DMA operation uses this mode.
3. Last Output Data Mode (LOD).
   This mode is used to generate message authentication code (MAC) on data in CCM mode of operation. The CCM mode combines counter mode for encryption and CBC-MAC generation for authentication.

When LOD is disabled in CCM mode then counter mode of encryption is performed on the input data block.

When LOD is enabled in CCM mode then CBC-MAC generation is performed. Zero block is used as the initialization vector by the hardware. Reading from the Output Data Register (INDATA) is not required to clear the ENCCMP flag. The ENCCMP flag is automatically cleared by writing to the Input Data Register (INDATA). This allows retrieval of only the last data in several encryption/decryption processes. No output data register reads are necessary between each block of encryption/decryption process.

Note that assembling message depending on the security level identifier in CCM* has to be done in software.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 42.6.2.5. Computation of Last Nk Words of Expanded Key

The AES algorithm takes the cryptographic key provided by the user and performs a Key Expansion routine to generate an expanded key. The expanded key contains a total of 4(Nr + 1) 32-bit words, where the first Nk (4/6/8 for a 128/192/256-bit key) words are the user-provided key. For data encryption, the expanded key is used in the forward direction, that is, the first four words are used in the initial round of data processing, the second four words in the first round, the third four words in the second round, and so on. On the other hand, for data decryption, the expanded key is used in the reverse direction, i.e.,the last four words are used in the initial round of data processing, the last second four words in the first round, the last third four words in the second round, and so on.

To reduce gate count, the AES module does not generate and store the entire expanded key prior to data processing. Instead, it computes on-the-fly the round key (four 32-bit words) required for the current round of data processing. In general, the round key for the current round of data processing can be computed from the Nk words of the expanded key generated in the previous rounds. When AES module is operating in the encryption mode, the round key for the initial round of data processing is simply the user-provided key written to the KEY registers. On the other hand, when AES module is operating in the decryption mode, the round key for the initial round of data processing is the last four words of the expanded key, which is not available unless AES module has performed at least one encryption process prior to operating in the decryption mode.

In general, the last Nk words of the expanded key must be available before decryption can start. If desired, AES module can be instructed to compute the last Nk words of the expanded key in advance by writing a one to the Key Generate (KEYGEN) bit in the CTRLA register (CTRLA.KEYGEN). The computation takes Nr clock cycles. Alternatively, the last Nk words of the expanded key can be automatically computed by AES module when a decryption process is initiated if they have not been computed in advance or have become invalid. Note that this will introduce a latency of Nr clock cycles to the first decryption process.

#### 42.6.2.6. Hardware Countermeasures against Differential Power Analysis Attacks

The AES module features the following four types of hardware countermeasures that are useful for protecting data against differential power analysis attacks:

- Type 1: Randomly add one cycle to data processing
- Type 2: Randomly add one cycle to data processing (other version)
- Type 3: Add a random number of clock cycles to data processing, subject to a maximum of 11/13/15 clock cycles for key sizes of 128/192/256 bits
- Type 4: Add random spurious power consumption during data processing

By default, all countermeasures are enabled, but require a write in DRNGSEED register to be effective. One or more of the countermeasures can be disabled by programming the Countermeasure Type field in the Control A (CTRLA.CTYPE) register. The countermeasures use random numbers generated by a deterministic random number generator embedded in AES module. The seed for the random number generator is written to the RANDSEED register. Also, a new seed must be written after a change in the key size. Note that enabling countermeasures reduces AES module's throughput. In short, the throughput is highest with all the countermeasures disabled. On the other hand, with all of the countermeasures enabled, the best protection is achieved but the throughput is worst.

### 42.6.3. Galois Counter Mode (GCM)

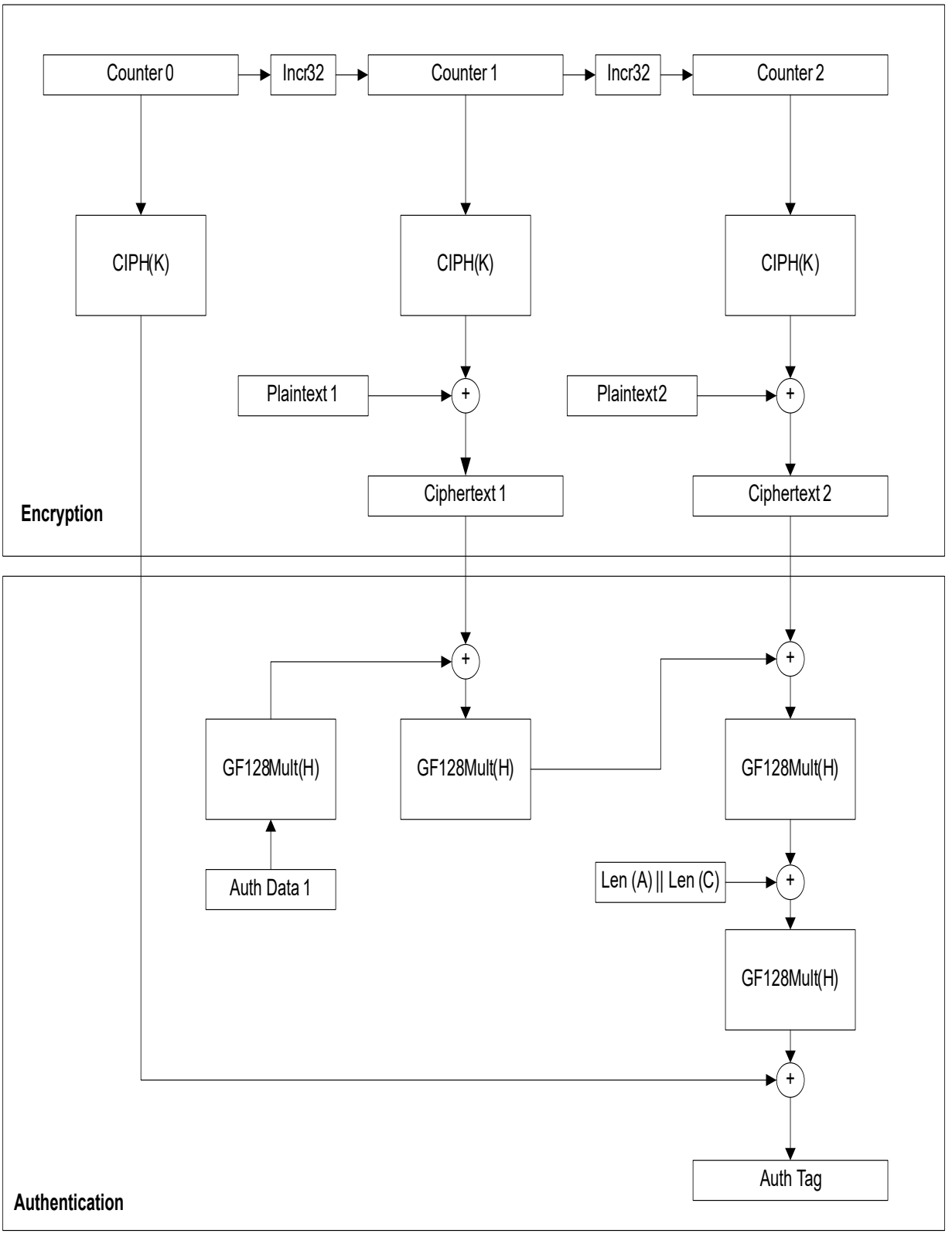
GCM is comprised of the AES engine in CTR mode along with a universal hash function (GHASH engine) that is defined over a binary Galois field to produce a message authentication tag. The GHASH engine processes data packets after the AES operation. GCM provides assurance of the confidentiality of data through the AES Counter mode of operation for encryption, see figure below. Authenticity of the confidential data is assured through the GHASH engine. Refer to the "NIST Special Publication 800-38D Recommendation" for additional information.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

GCM processing is computed on 128-bit input data fields. There is no support for unaligned data. The AES key length can be any length that is supported by the AES module.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 42.6.3.1. GCM Operation

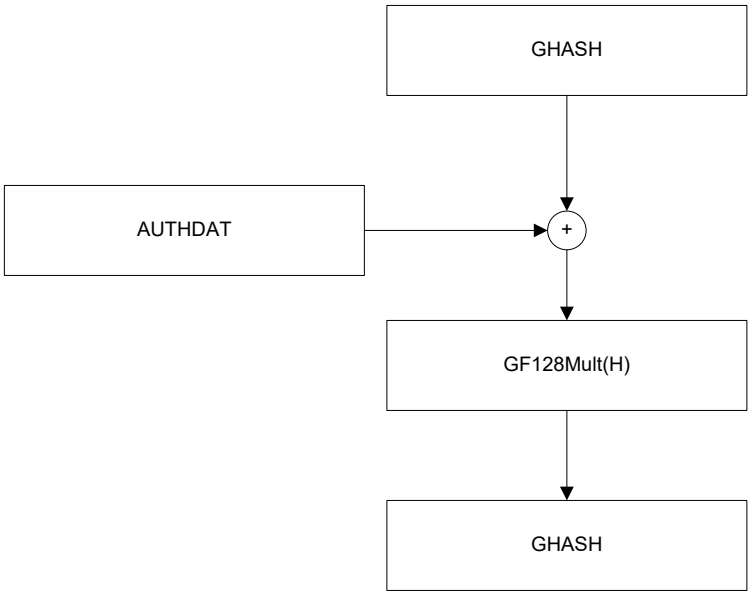
##### 42.6.3.1.1. Hash Key Generation

1. Configure the CTRLA register as follows:
   a. CTRLA.STARTMODE as manual (Auto for DMAC).
   b. CTRLA.CIPHER as encryption.
   c. CTRLA.KEYSIZE as per the key used.
   d. CTRLA.AESMODE as ECB.
   e. CTRLA.CTYPE as per the countermeasures required.
2. Set CTRLA.ENABLE.
3. Write zero to the CIPLEN register.
4. Write the key in the KEYWORD register.
5. Write the zeros to the INDATA register.
6. Set CTRLB.Start.
7. Wait for INTFLAG.ENCCMP to be set.
8. AES Hardware generates Hash Subkey in the HASHKEY register.

##### 42.6.3.1.2. Authentication Header Processing

- Configure CTRLA register as follows:
  a. CTRLA.STARTMODE as Manual
  b. CTRLA.CIPHER as Encryption
  c. CTRLA.KEYSIZE as per the key used
  d. CTRLA.AESMODE as GCM
  e. CTRLA.CTYPE as per the countermeasures required.
- Set CTRLA.ENABLE
- Write the key in KEYWORD register
- Set CTRLB.GFMUL
- Write the Authdata to INDATA reg
- Set CTRLB.START as1
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates output in GHASH register
- Continue steps 4 to 7 for remaining Authentication Header.
  Note: If the Auth data is less than 128 bit, it has to be padded with zero to make it 128 bit aligned.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.




#### 42.6.3.1.3. Plain text Processing

- Set CTRLB.NEWMSG for the new set of plain text processing.
- Load CIPLEN reg.
- Load (J0+1) in INTVECT register.
- As described in NIST documentation J 0 = IV || 0 31 || 1 when len(IV)=96 and J0 =$GHASH_H$ (IV || 0 s+64 || [len(IV)] 64 ) (s is the minimum number of zeroes that should be padded with the Initialization Vector to make it a multiple of 128) if len(IV) != 96.
- Load plain text in INDATA register.
- Set CTRLB.START as 1.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in INDATA register.
- Intermediate GHASH is stored in GHASH register and Cipher Text available in INDATA register.
- Continue 3 to 6 till the input of plain text to get the cipher text and the Hash keys.
- At the last input, set CTRLB.EOM.
- Write last in-data to INDATA reg.
- Set CTRLB.START as 1.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in INDATA register and final Hash key in GHASH register.
- Load [LEN(A)]64||[LEN(C)]64 in INDATA register and set CTRLB.GFMUL and CTRLB.START as 1.
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates final GHASH value in GHASH register.

#### 42.6.3.1.4. Plain text processing with DMAC

- Set CTRLB.NEWMSG for the new set of plain text processing.
- Load CIPLEN reg.
- Load (J0+1) in INTVECT register.
- Load plain text in INDATA register.
- Wait for INTFLAG.ENCCMP to be set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- AES Hardware generates output in INDATA register.
- Intermediate GHASH is stored in GHASH register and Cipher Text available in INDATA register.
- Continue 3 to 5 till the input of plain text to get the cipher text and the Hash keys.
- At the last input, set CTRLB.EOM.
- Write last in-data to INDATA reg.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates output in INDATA register and final Hash key in GHASH register.
- Load [LEN(A)]64||[LEN(C)]64 in INDATA register and set CTRLB.GFMUL and CTRLB.START as 1.
- Wait for INTFLAG.GFMCMP to be set.
- AES Hardware generates final GHASH value in GHASH register.

#### 42.6.3.1.5. Tag Generation

- Configure CTRLA
  a. Set CTRLA.ENABLE to 0
  b. Set CTRLA.AESMODE as CTR
  c. Set CTRLA.ENABLE to 1
- Load J0 value to INITVECTV reg.
- Load GHASH value to INDATA reg.
- Set CTRLB.NEWMSG and CTRLB.START to start the Counter mode operation.
- Wait for INTFLAG.ENCCMP to be set.
- AES Hardware generates the GCM Tag output in INDATA register.

### 42.6.4. Synchronization

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 42.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | CFBS[2:0] | | | AESMODE[2:0] | | ENABLE | SWRST |
| | | 15:8 | | XORKEY | KEYGEN | LOD | STARTMODE | CIPHER | KEYSIZE[1:0] | |
| | | 23:16 | | | | | CTYPE[3:0] | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CTRLB | 7:0 | | | | | GFMUL | EOM | NEWMSG | START |
| 0x05 | INTENCLR | 7:0 | | | | | | | GFMCMP | ENCCMP |
| 0x06 | INTENSET | 7:0 | | | | | | | GFMCMP | ENCCMP |
| 0x07 | INTFLAG | 7:0 | | | | | | | GFMCMP | ENCCMP |
| 0x08 | DATABUFPTR | 7:0 | | | | | | | INDATAPTR[1:0] | |
| 0x09 | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0A ... 0x0B | Reserved | | | | | | | | | |
| 0C | KEYWORD0 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 10 | KEYWORD1 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 14 | KEYWORD2 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 18 | KEYWORD3 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 1C | KEYWORD4 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 20 | KEYWORD5 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 24 | KEYWORD6 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 28 | KEYWORD7 | 7:0 | KEYWORD[7:0] | | | | | | | |
| | | 15:8 | KEYWORD[15:8] | | | | | | | |
| | | 23:16 | KEYWORD[23:16] | | | | | | | |
| | | 31:24 | KEYWORD[31:24] | | | | | | | |
| 0x2C ... 0x37 | Reserved | | | | | | | | | |
| 0x38 | INDATA | 7:0 | INDATA[7:0] | | | | | | | |
| | | 15:8 | INDATA[15:8] | | | | | | | |
| | | 23:16 | INDATA[23:16] | | | | | | | |
| | | 31:24 | INDATA[31:24] | | | | | | | |
| 3C | INTVECTV0 | 7:0 | INTVECTV[7:0] | | | | | | | |
| | | 15:8 | INTVECTV[15:8] | | | | | | | |
| | | 23:16 | INTVECTV[23:16] | | | | | | | |
| | | 31:24 | INTVECTV[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 40 | INTVECTV1 | 7:0 | INTVECTV[7:0] | | | | | | | |
| | | 15:8 | INTVECTV[15:8] | | | | | | | |
| | | 23:16 | INTVECTV[23:16] | | | | | | | |
| | | 31:24 | INTVECTV[31:24] | | | | | | | |
| 44 | INTVECTV2 | 7:0 | INTVECTV[7:0] | | | | | | | |
| | | 15:8 | INTVECTV[15:8] | | | | | | | |
| | | 23:16 | INTVECTV[23:16] | | | | | | | |
| | | 31:24 | INTVECTV[31:24] | | | | | | | |
| 48 | INTVECTV3 | 7:0 | INTVECTV[7:0] | | | | | | | |
| | | 15:8 | INTVECTV[15:8] | | | | | | | |
| | | 23:16 | INTVECTV[23:16] | | | | | | | |
| | | 31:24 | INTVECTV[31:24] | | | | | | | |
| 0x4C ... 0x5B | Reserved | | | | | | | | | |
| 0x5C | HASHKEY0 | 7:0 | HASHKEY[7:0] | | | | | | | |
| | | 15:8 | HASHKEY[15:8] | | | | | | | |
| | | 23:16 | HASHKEY[23:16] | | | | | | | |
| | | 31:24 | HASHKEY[31:24] | | | | | | | |
| 0x60 | HASHKEY1 | 7:0 | HASHKEY[7:0] | | | | | | | |
| | | 15:8 | HASHKEY[15:8] | | | | | | | |
| | | 23:16 | HASHKEY[23:16] | | | | | | | |
| | | 31:24 | HASHKEY[31:24] | | | | | | | |
| 0x64 | HASHKEY2 | 7:0 | HASHKEY[7:0] | | | | | | | |
| | | 15:8 | HASHKEY[15:8] | | | | | | | |
| | | 23:16 | HASHKEY[23:16] | | | | | | | |
| | | 31:24 | HASHKEY[31:24] | | | | | | | |
| 0x68 | HASHKEY3 | 7:0 | HASHKEY[7:0] | | | | | | | |
| | | 15:8 | HASHKEY[15:8] | | | | | | | |
| | | 23:16 | HASHKEY[23:16] | | | | | | | |
| | | 31:24 | HASHKEY[31:24] | | | | | | | |
| 0x6C | GHASH0 | 7:0 | GHASH[7:0] | | | | | | | |
| | | 15:8 | GHASH[15:8] | | | | | | | |
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x70 | GHASH1 | 7:0 | GHASH[7:0] | | | | | | | |
| | | 15:8 | GHASH[15:8] | | | | | | | |
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x74 | GHASH2 | 7:0 | GHASH[7:0] | | | | | | | |
| | | 15:8 | GHASH[15:8] | | | | | | | |
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x78 | GHASH3 | 7:0 | GHASH[7:0] | | | | | | | |
| | | 15:8 | GHASH[15:8] | | | | | | | |
| | | 23:16 | GHASH[23:16] | | | | | | | |
| | | 31:24 | GHASH[31:24] | | | | | | | |
| 0x7C ... 0x7F | Reserved | | | | | | | | | |
| 0x80 | CIPLEN | 7:0 | CIPLEN[7:0] | | | | | | | |
| | | 15:8 | CIPLEN[15:8] | | | | | | | |
| | | 23:16 | CIPLEN[23:16] | | | | | | | |
| | | 31:24 | CIPLEN[31:24] | | | | | | | |
| 0x84 | RANDSEED | 7:0 | RANDSEED[7:0] | | | | | | | |
| | | 15:8 | RANDSEED[15:8] | | | | | | | |
| | | 23:16 | RANDSEED[23:16] | | | | | | | |
| | | 31:24 | RANDSEED[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 42.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 42.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CTYPE[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | XORKEY | KEYGEN | LOD | STARTMODE | CIPHER | KEYSIZE[1:0] | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CFBS[2:0] | | | AESMODE[2:0] | | | ENABLE | SWRST |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 19:16 – CTYPE[3:0]** Counter Measure Type

| Value | Name | Description |
|---|---|---|
| XXX0 | CTYPE1 disabled | Countermeasure1 disabled |
| XXX1 | CTYPE1 enabled | Countermeasure1 enabled |
| XX0X | CTYPE2 disabled | Countermeasure2 disabled |
| XX1X | CTYPE2 enabled | Countermeasure2 enabled |
| X0XX | CTYPE3 disabled | Countermeasure3 disabled |
| X1XX | CTYPE3 enabled | Countermeasure3 enabled |
| 0XXX | CTYPE4 disabled | Countermeasure4 disabled |
| 1XXX | CTYPE4 enabled | Countermeasure4 enabled |

**Bit 14 – XORKEY** XOR Key Operation

| Value | Description |
|---|---|
| 0 | No effect |
| 1 | The user keyword gets XORed with the previous keyword register content. |

**Bit 13 – KEYGEN** Last Key Generation

| Value | Description |
|---|---|
| 0 | No effect |
| 1 | Start Computation of the last NK words of the expanded key |

**Bit 12 – LOD** Last Output Data Mode

| Value | Description |
|---|---|
| 0 | No effect |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Start encryption in Last Output Data mode |

**Bit 11 – STARTMODE** Start Mode Select

| Value | Name | Description |
|---|---|---|
| 0 | Manual Mode | Start Encryption / Decryption in Manual mode |
| 1 | Auto Mode | Start Encryption / Decryption in Auto mode |

**Bit 10 – CIPHER** Cipher Mode Select

| Value | Description |
|---|---|
| 0 | Decryption |
| 1 | Encryption |

**Bits 9:8 – KEYSIZE[1:0]** Encryption Key Size

| Value | Name | Description |
|---|---|---|
| 0 | 128-bit Key | 128-bit Key for Encryption / Decryption |
| 1 | 192-bit Key | 192-bit Key for Encryption / Decryption |
| 2 | 256-bit Key | 256-bit Key for Encryption / Decryption |
| 3 | Reserved | Reserved |

**Bits 7:5 – CFBS[2:0]** Cipher Feedback Block Size

| Value | Name | Description |
|---|---|---|
| 0 | 128-bit data block | 128-bit Input data block for Encryption/Decryption in Cipher Feedback mode |
| 1 | 64-bit data block | 64-bit Input data block for Encryption/Decryption in Cipher Feedback mode |
| 2 | 32-bit data block | 32-bit Input data block for Encryption/Decryption in Cipher Feedback mode |
| 3 | 16-bit data block | 16-bit Input data block for Encryption/Decryption in Cipher Feedback mode |
| 4 | 8-bit data block | 8-bit Input data block for Encryption/Decryption in Cipher Feedback mode |
| 5–7 | Reserved | Reserved |

**Bits 4:2 – AESMODE[2:0]** AES Modes of Operation

| Value | Name | Description |
|---|---|---|
| 0 | ECB | Electronic code book mode |
| 1 | CBC | Cipher block chaining mode |
| 2 | OFB | Output feedback mode |
| 3 | CFB | Cipher feedback mode |
| 4 | Counter | Counter mode |
| 5 | CCM | CCM mode |
| 6 | GCM | Galois counter mode |
| 7 | Reserved | Reserved |

**Bit 1 – ENABLE** Enable

| Value | Description |
|---|---|
| 0 | The peripheral is disabled |
| 1 | The peripheral is enabled |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the AES module to their initial state, and the module will be disabled.
Writing a '1' to `SWRST` will always take precedence, meaning that all other writes in the same write operation will be discarded.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing |
| 1 | The reset operation is ongoing |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | GFMUL | EOM | NEWMSG | START |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 3 – GFMUL** GF Multiplication
This bit is applicable only to GCM mode.

| Value | Description |
|---|---|
| 0 | No action |
| 1 | Setting this bit calculates GF multiplication with data buffer content and hashkey register content. |

**Bit 2 – EOM** End of Message
This bit is applicable only to GCM mode.

| Value | Description |
|---|---|
| 0 | No action |
| 1 | Setting this bit generates final GHASH value for the message. |

**Bit 1 – NEWMSG** New Message
This bit is used in cipher block chaining (CBC), cipher feedback (CFB) and output feedback (OFB), counter (CTR) modes to indicate the hardware to use Initialization vector for encrypting the first block of message.

| Value | Description |
|---|---|
| 0 | No action |
| 1 | Setting this bit indicates start of new message to the module. |

**Bit 0 – START** Start Encryption/Decryption

| Value | Description |
|---|---|
| 0 | No action |
| 1 | Start encryption / decryption in manual mode. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.3. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | GFMCMP | ENCCMP |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – GFMCMP** GF Multiplication Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the GF Multiplication Complete Interrupt Enable bit, which disables the GF Multiplication Complete interrupt.

| Value | Description |
|---|---|
| 0 | The GF Multiplication Complete interrupt is disabled. |
| 1 | The GF Multiplication Complete interrupt is enabled. |

**Bit 0 – ENCCMP** Encryption Complete Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Encryption Complete Interrupt Enable bit, which disables the Encryption Complete interrupt.

| Value | Description |
|---|---|
| 0 | The Encryption Complete interrupt is disabled. |
| 1 | The Encryption Complete interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.4. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x06
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | GFMCMP | ENCCMP |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – GFMCMP** GF Multiplication Complete Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the GF Multiplication Complete Interrupt Enable bit, which enables the GF Multiplication Complete interrupt.

| Value | Description |
|---|---|
| 0 | The GF Multiplication Complete interrupt is disabled. |
| 1 | The GF Multiplication Complete interrupt is enabled. |

**Bit 0 – ENCCMP** Encryption Complete Interrupt Enable
Writing a '0' to this bit has no effect. Writing a '1' to this bit will clear the Encryption Complete Interrupt Enable bit, which enables the Encryption Complete interrupt.

| Value | Description |
|---|---|
| 0 | The Encryption Complete interrupt is disabled. |
| 1 | The Encryption Complete interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.5. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x07
**Reset:** 0x00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | GFMCMP | ENCCMP |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – GFMCMP** GF Multiplication Complete
This flag is cleared by writing a '1' to it.
This flag is set when GHASH value is available on the Galois Hash Registers (GHASHx) in GCM mode.
Writing a '0' to this bit has no effect.
This flag is also automatically cleared in the following cases.

1. Manual encryption/decryption occurs (START in CTRLB register).
2. Reading from the GHASHx register.

**Bit 0 – ENCCMP** Encryption Complete
This flag is cleared by writing a '1' to it.
This flag is set when encryption/decryption is complete and valid data is available on the Data Register.
Writing a '0' to this bit has no effect.
This flag is also automatically cleared in the following cases:

1. Manual encryption/decryption occurs (START in CTRLA register). (This feature is needed only if we do not support double buffering of INDATA registers).
2. Reading from the data register (INDATAx) when LOD = 0.
3. Writing into the data register (INDATAx) when LOD = 1.
4. Reading from the Hash Key register (HASHKEYx).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.6. Data Buffer Pointer

**Name:** DATABUFPTR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | INDATAPTR[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – INDATAPTR[1:0]** Input Data Pointer
Writing to this field changes the value of the input data pointer, which determines which of the four data registers is written to/read from when the next write/read to the `INDATA` register address is performed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.7. Debug

**Name:** DBGCTRL
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
Writing a '0' to this bit causes the AES to halt during debug mode.
Writing a '1' to this bit allows the AES to continue normal operation during debug mode. This bit can only be changed while the AES is disabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.8. Keyword

**Name:** KEYWORD
**Offset:** 0x0C + n*0x04 [n=0..7]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | KEYWORD[31:24] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | KEYWORD[23:16] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | KEYWORD[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | KEYWORD[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – KEYWORD[31:0]** Key Word Value

The four/six/eight 32-bit Key Word registers set the 128-bit/192-bit/256-bit cryptographic key used for encryption/decryption. `KEYWORD0.KEYWORD` corresponds to the first word of the key and `KEYWORD3/KEYWORD5/KEYWORD7.KEYWORD` to the last one.

**Note:** By setting the XORKEY bit of CTRLA register, keyword will update with the resulting XOR value of user keyword and previous keyword content.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.9. Data

**Name:** INDATA
**Offset:** 0x38
**Reset:** 0x00000000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | INDATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | INDATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | INDATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | INDATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – INDATA[31:0]** Data Value

A write to or read from this register corresponds to a write to or read from one of the four data registers. The four 32-bit Data registers set the 128-bit data block used for encryption/decryption. The data register that is written to or read from is given by the `DATABUFPTR.INDATPTR` field.

**Note:** Both input and output shares the same data buffer. Reading INDATA register will return 0's when AES is performing encryption or decryption operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.10. Initialization Vector Register

**Name:** INTVECTV
**Offset:** 0x3C + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | INTVECTV[31:24] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | INTVECTV[23:16] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | INTVECTV[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | INTVECTV[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – INTVECTV[31:0]** Initialization Vector Value

The four 32-bit Initialization Vector registers `INTVECTVn` set the 128-bit Initialization Vector data block that is used by some modes of operation as an additional initial input. `INTVECTV0.INTVECTV` corresponds to the first word of the Initialization Vector, `INTVECTV3.INTVECTV` to the last one. These registers are write-only to prevent the Initialization Vector from being read by another application. For CBC, OFB, and CFB modes, the Initialization Vector corresponds to the initialization vector. For CTR mode, it corresponds to the counter value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.11. Hash Key (GCM mode only)

**Name:** HASHKEY
**Offset:** 0x5C + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** PAC Write-protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | HASHKEY[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | HASHKEY[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | HASHKEY[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | HASHKEY[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – HASHKEY[31:0]** Hash Key Value

The four 32-bit HASHKEY registers contain the 128-bit Hash Key value computed from the AES KEY. The Hash Key value can also be programmed offering single GF128 multiplication possibilities.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.12. Galois Hash (GCM mode only)

**Name:** GHASH
**Offset:** 0x6C + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | GHASH[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | GHASH[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | GHASH[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GHASH[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – GHASH[31:0]** Galois Hash Value

The four 32-bit Hash Word registers GHASHcontain the GHASH value after GF128 multiplication in GCM mode. Writing a new key to KEYWORD registers causes GHASH to be initialized with zeroes. These registers can also be programmed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.13. Galois Hash x (GCM mode only)

**Name:** CIPLEN
**Offset:** 0x80
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CIPLEN[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CIPLEN[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CIPLEN[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CIPLEN[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CIPLEN[31:0]** Cipher Length
This register contains the length in bytes of the Cipher text that is to be processed. This is programmed by the user in GCM mode for Tag generation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.8.14. Random Seed

**Name:** RANDSEED
**Offset:** 0x84
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RANDSEED[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RANDSEED[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RANDSEED[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RANDSEED[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RANDSEED[31:0]** Random Seed
A write to this register corresponds to loading a new seed into the Random number generator.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 43. Public Key Cryptography Controller (PUKCC)

## 43.1. Overview

The Public Key Cryptography Controller (PUKCC) processes public key cryptography algorithm calculus in both GF(p) and GF(2n) fields.

The Public Key Cryptography Library (PUKCL) is stored in ROM inside the device. The library can be used in applications to access features of PUKCC, and includes the complete implementation of the following public key cryptography algorithms:

- RSA (Rivest-Shamir-Adleman public key cryptosystem), DSA (Digital Signature Algorithm):
  - Modular Exponentiation with CRT up to 7168 bits
  - Modular Exponentiation without CRT up to 5376 bits
  - Prime generation
  - Utilities: GCD/modular Inverse, Divide, Modular reduction, Multiply, ...
- Elliptic Curves:
  - ECDSA GF(p) up to 521 bits for common curves (up to 1120 bits for future use)
  - ECDSA GF(2n) up to 571 bits for common curves (up to 1440 bits for future use)
  - Choice of the curve parameters for compatibility with NIST Curves or other curves in Weierstrass equation
  - Point Multiply
  - Point Add/Doubling
  - Other high level elliptic curve algorithms (ECDH, ...) can be implemented by user using library functions
- Deterministic Random Number Generation (DRNG ANSI X9.31) for DSA

## 43.2. Product Dependencies

### 43.2.1. I/O Lines

Not applicable.

### 43.2.2. Power Management

The PUKCC will continue to operate in any sleep mode, as long as its source clock is running.

### 43.2.3. Clocks

The bus clock (CLK_PUKCC_AHB) can be enabled and disabled by the Main Clock Controller.

**Related Links**

MCLK – Main Clock

### 43.2.4. DMA

Not applicable.

### 43.2.5. Interrupts

Not applicable.

### 43.2.6. Events

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 43.3. Functional Description

### 43.3.1. Public Key Cryptography Library (PUKCL) Application Programming Interface (API)

The Public Key Cryptography Controller (PUKCC) is a peripheral that can be used to accelerate public key cryptography, and processes public key cryptography algorithm calculus in both Prime field (GF(p)) and Binary field (GF($2^n$)). Different functionalities of the PUKCC are accessed with the help of the Public Key Cryptography Library (PUKCL), which is embedded into a dedicated ROM inside the microcontroller.

The PUKCL provides access to many algorithms and functions. The features provided, start from basic addition or comparison, up to the RSA or ECDSA complete computation. The library can be utilized by including the PUKCL Driver in the application and passing parameters through the Application Programming Interface (API). The PUKCC Driver is available in Atmel START within *Drivers* > *Cryptography*. This library can be used in conjunction with a SSL software stack to improve performance and helps to reduce the RAM usage and time taken to perform different cryptographic functions.

### 43.3.2. PUKCL Features

PUKCL features include:

- Basic Arithmetic and Cryptographic Services - PUKCL self-test, GCD, integral division, etc.
- Modular Arithmetic Services - Modular reduction, modular exponentiation, probable prime generation and modular exponentiation
- Elliptic Curves Over GF(p) Services - Point addition and doubling on an elliptic curve in a prime field, ECDSA signature generation and verification on an elliptic curve over GF(p)
- Elliptic Curves Over GF(2n) Services - Point addition and doubling on an elliptic curve in a prime field, ECDSA signature generation and verification on an elliptic curve over GF($2^n$)

### 43.3.3. PUKCL Usage

The following sections provide details on accessing the PUKCL and its features.

#### 43.3.3.1. Initializing the PUKCC and PUKCL

For a project created with Atmel START, the clock initialization is handled by the initialization function `atmel_start_init`. After a power-on reset, and when the PUKCC Clock is enabled, a Crypto RAM clear process is launched. It is mandatory to wait until the end of this process before using the Crypto Library.

The following code shows how to wait for the Crypto RAM clear process.

```
while ((PUKCCSR & BIT_PUKCCSR_CLRRAM_BUSY) != 0);
```

The next task to be done is self-test. From the generated project in Atmel Studio, copy the example for the PUKCC Driver SelfTest and add it to the main source file. This is a mandatory step before using the library. The return values from the SelfTest service must be compared against known values mentioned in the service description (see the **Description** section in SelfTest).

**Example 43-1.** PUKCC Initialization

```
void PUKCC_self_test(void)
{
    // Clear contents of PUKCLParam
    memset(&PUKCLParam, 0, sizeof(PUKCL_PARAM));

    pvPUKCLParam = &PUKCLParam;
    vPUKCL_Process(SelfTest, pvPUKCLParam);

    // In case of error, loop here
    while (PUKCL(u2Status) != PUKCL_OK) {
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
    ;
    }
    while (pvPUKCLParam->P.PUKCL_SelfTest.u4Version != PUKCL_VERSION) {
    ;
    }
    while (pvPUKCLParam->P.PUKCL_SelfTest.u4CheckNum1 != 0x6E70DDD2) {
    ;
    }
    while (pvPUKCLParam->P.PUKCL_SelfTest.u4CheckNum2 != 0x25C8D64F) {
    ;
    }
}

int main(void)
{
    /* Initializes MCU, drivers and middleware */
    atmel_start_init();

    // Wait for Crypto RAM clear process
    while ((PUKCCSR & BIT_PUKCCSR_CLRRAM_BUSY) != 0);

    // Initialize PUKCC and perform self test
    PUKCC_self_test();
    while(1)
    {
    }
}
```

**Note:** It may also be necessary to initialize the Random Number Generator (RNG) on the microcontroller, as some services in the library use the peripheral. Before calling such services, be sure to follow the directives given for random number generation on the selected microcontroller (particularly initialization and seeding) and compulsorily start the RNG. For details refer to each service.

### 43.3.3.2. Accessing Different Library Services

All cryptographic services in the library are accessed by the macro vPUKCL_Process. All of these services use the same process for receiving and returning parameters. PUKCL receives two arguments: the requested service and a pointer to a structure called the parameter block. The parameter block contains two structures, a common parameter structure for all commands and specific parameter structure for each service. A specific service is accessed with vPUKCL_Process by passing the service name as the first argument. For example, to perform SelfTest, use vPUKCL_Process(SelfTest, pvPUKCLParam).

**Example 43-2.** PUKCL Parameter Block

```
typedef struct _PUKCL_param {
    PUKCL_HEADER PUKCL_Header;
    union {
    _PUKCL_CLEARFLAGS PUKCL_ClearFlags;
    _PUKCL_COMP       PUKCL_Comp;
    _PUKCL_CONDCOPY   PUKCL_CondCopy;
    _PUKCL_CRT        PUKCL_CRT;
    _PUKCL_DIV        PUKCL_Div;
    _PUKCL_EXPMOD     PUKCL_ExpMod;
    _PUKCL_FASTCOPY   PUKCL_FastCopy;
    _PUKCL_FILL       PUKCL_Fill;
    _PUKCL_FMULT      PUKCL_Fmult;
    _PUKCL_GCD        PUKCL_GCD;
    _PUKCL_PRIMEGEN   PUKCL_PrimeGen;
    _PUKCL_REDMOD     PUKCL_RedMod;
    _PUKCL_RNG        PUKCL_Rng;
    _PUKCL_SELFTEST   PUKCL_SelfTest;
    _PUKCL_SMULT      PUKCL_Smult;
    _PUKCL_SQUARE     PUKCL_Square;
    _PUKCL_SWAP       PUKCL_Swap;

    // ECC
    _PUKCL_ZPECCADD                 PUKCL_ZpEccAdd;
    _PUKCL_ZPECCDBL                 PUKCL_ZpEccDbl;
    _PUKCL_ZPECCADDSUB              PUKCL_ZpEccAddSub;
    _PUKCL_ZPECCMUL                 PUKCL_ZpEccMul;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
    _PUKCL_ZPECDSAGENERATE             PUKCL_ZpEcDsaGenerate;
    _PUKCL_ZPECDSAVERIFY               PUKCL_ZpEcDsaVerify;
    _PUKCL_ZPECDSAQUICKVERIFY          PUKCL_ZpEcDsaQuickVerify;
    _PUKCL_ZPECCQUICKDUALMUL           PUKCL_ZpEccQuickDualMul;
    _PUKCL_ZPECCONVPROJTOAFFINE        PUKCL_ZpEcConvProjToAffine;
    _PUKCL_ZPECCONVAFFINETOPROJECTIVE PUKCL_ZpEcConvAffineToProjective;
    _PUKCL_ZPECRANDOMIZECOORDINATE     PUKCL_ZpEcRandomiseCoordinate;
    _PUKCL_ZPECPOINTISONCURVE          PUKCL_ZpEcPointIsOnCurve;

    // ECC
    _PUKCL_GF2NECCADD                    PUKCL_GF2NEccAdd;
    _PUKCL_GF2NECCDBL                    PUKCL_GF2NEccDbl;
    _PUKCL_GF2NECCMUL                    PUKCL_GF2NEccMul;
    _PUKCL_GF2NECDSAGENERATE             PUKCL_GF2NEcDsaGenerate;
    _PUKCL_GF2NECDSAVERIFY               PUKCL_GF2NEcDsaVerify;
    _PUKCL_GF2NECCONVPROJTOAFFINE        PUKCL_GF2NEcConvProjToAffine;
    _PUKCL_GF2NECCONVAFFINETOPROJECTIVE PUKCL_GF2NEcConvAffineToProjective;
    _PUKCL_GF2NECRANDOMIZECOORDINATE     PUKCL_GF2NEcRandomiseCoordinate;
    _PUKCL_GF2NECPOINTISONCURVE          PUKCL_GF2NEcPointIsOnCurve;
    } P;
} PUKCL_PARAM,
```

#### 43.3.3.2.1. PUKCL_HEADER Structure

The PUKCL_HEADER is common for all services of the library. This header includes standard fields to indicate the requested service, sub-service, options, return status, and so on, as shown in the following tables.

Different terms used in the below description to be understood, are as follows:

- Parameter – Represents a variable used by the PUKCL. Every parameter belongs to either PUKCL_HEADER or PUKCL Service Specific Header
- Type – Indicates the data type. For details on data type, please refer to `CryptoLib_typedef_pb.h` file in the library
- Dir – Direction. Indicates whether PUKCL considers the variable as input or output. Input means that the application passes data to the PUKCL using the variable. Output means that the PUKCL uses the variable to pass data to the application.
- Location – Suggests whether the parameter need to be stored in Crypto RAM or device SRAM. The PUKCL driver has macros for placing parameters into Crypto RAM, so that the user does not have to worry about the addresses
- Data Length – If a parameter is a pointer variable, the Data Length column shows the size of the data pointed by the pointer

**Table 43-1.** PUKCL_HEADER Structure

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u1Service | u1 | I | – | – | Required service | Executed service |
| u1SubService | u1 | I | – | – | Required sub-service | Executed sub-service |
| u2Option | u2 | I | – | – | Required option | Executed option |
| Specific | PUKCL_STATUS | I/O | – | – | See Table 43-2 | See Table 43-2 |
| u2Status (see Return Codes) | u2 | I/O | – | – | – | Output Status |
| Reserved | u2 | – | – | – | – | – |
| Reserved | u4 | – | – | – | – | – |

The Specific field in the PUKCL_HEADER structure is another structure named PUKCL_STATUS. The following table describes this structure. The details of the use of these bits are provided in the individual service descriptions.

#### 43.3.3.2.2. PUKCL_STATUS Structure

Members of the PUKCL_STATUS structure are shown in the following table.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-2.** PUKCL_STATUS Structure

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| CarryIn (see **Note 1**) | bit | I | – | – | CarryIn | – |
| CarryOut | bit | O | – | – | – | CarryOut |
| Zero | bit | O | – | – | – | 1: Result is zero<br>0: Result is not zero |
| Gf2n (see **Note 1**) | bit | I | – | – | Mathematical field 0: Integers ($Z_p$)<br>1: Field GF($2^n$) | – |
| Violation | bit | O | – | – | – | Indicates a violation |

**Note:**

1. Two of these fields must be filled in to avoid problems during computations. If the Gf2n and CarryIn fields are not reset or initialized properly, problems may be encountered during computations. For instance, not initializing the Gf2n field may result in getting a correct mathematical result, but computed over GF($2^n$) instead of $Z_p$.

#### 43.3.3.2.3. PUKCL Service Specific Header

Details about each service specific header are provided with service descriptions in a subsequent section. Such structures may contain input or output parameters. A parameter is considered as an input parameter when it used for passing information to the PUKCL, and it is considered as an output parameter when the PUKCL uses it to pass a result back to the application code.

The following code provides the service specific header example for the SelfTest service.

```
typedef struct _PUKCL_selftest {
    u4 u4Version;
    u4 u4PUKCCVersion;
    u4 u4CheckNum1;
    u4 u4CheckNum2;
    u1 u1Step;
} _PUKCL_SELFTEST;
```

After the SelfTest service is invoked (with vPUKCL_Process(SelfTest, pvPUKCLParam)), the service specific return values can be checked using pvPUKCLParam.

To check whether the version returned by the PUKCL is correct, the following code can be used.

```
while (pvPUKCLParam->P.PUKCL_SelfTest.u4Version != PUKCL_VERSION);
```

In a similar way, other returns can also be accessed.

### 43.3.3.3. Parameter Passing (Special Considerations)

Most of the PUKCL services work with memory area and accept pointers and lengths as parameters to define input and output areas. Most of the time, the pointers and lengths are untouched by the services, while the defined areas are read, filled, or overwritten. These memory areas are defined with an initial pointer and a byte length. For most of the commands, the memory area location must be in the PUKCC Cryptographic RAM. The Cryptographic RAM is the memory area for parameter exchange with the PUKCL and is 4 Kbytes large. Sometimes memory areas can be located in Embedded SRAM, which is detailed in the Location column of the parameters description tables.

When working with binary fields, polynomials in GF($2^n$) need no transformation to be written in an area:

- Each bit represents a polynomial coefficient 0 or 1
- The polynomials must be written Low Significant Byte First
- A zero padding on the Most Significant Bytes may be added if the area is larger than the real size of the polynomial


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Important:** The Cryptographic RAM is 4 Kbytes in size and is dedicated to PUKCC. However, to ensure correct library operation, the two last 32-bit words must not be used. Unless otherwise specified, these memory areas contain integers in GF(p) or polynomials in GF($2^n$) with the Less Significant Byte first.

Unless otherwise specified, the length must be a multiple of four and the pointers must be four bytes aligned. This is because most of the services work with 32-bit words.

#### 43.3.3.4. Aligned Significant Length

Parameters in memory areas can have any Significant Length in bytes. As the lengths in PUKCL must be a multiple of four, a padding is processed on the Most Significant Side with zero to three bytes cleared to zero. Now the parameter can be considered to meet the Aligned Significant Length requirement for PUKCL.

#### 43.3.3.5. Processing Field GF(p) and GF($2^n$)

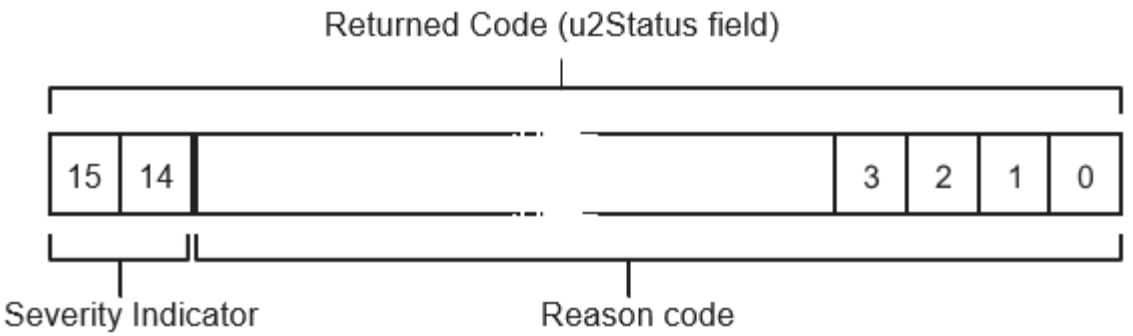
The library can process arithmetic functions over GF(p) (or Zp integers) and GF($2^n$), when applicable. The choice of these processing fields is made using the following rules:

- If a processing field is not applicable to the function, it is not mentioned and the Specific.GF2n bit has no effect.
- If the function can support both processing fields, the choice is mentioned and the Specific.GF2n bit must be filled according to the choice.
- If the function supports only one of the processing fields, the processing field is mentioned and the Specific.GF2n bit has no effect.

#### 43.3.3.6. Return Codes

Each call to one of the PUKCL services returns a status code indicating whether or not the execution is correct, which can be decoded, as shown in the following figure.

**Figure 43-1.** Return Code Status Decoding




The following table shows how the severity indicators should be decoded.

**Table 43-3.** Severity Indicators

| Value for Bits 14–15 | Severity | Comment |
|---|---|---|
| 0xC000 | Severe | Indicates a blocking error condition |
| 0x8000 | Warning | Indicates a cautionary use of the return values |
| 0x4000 | Information | Indicates the result is correct and gives information |
| 0x0000 | – | No error or no severity given |

The following table contains the exhaustive list of all reason codes.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-4.** Return Codes

| Value for Bits 00–13 | Severity Code | Reason Code |
|---|---|---|
| 0x0000 | – | PUKCL_OK |
| 0x4001 | Informative | PUKCL_NUMBER_IS_NOT_PRIME |
| 0x4002 | Informative | PUKCL_NUMBER_IS_PRIME |
| 0xC001 | Severe | PUKCL_COMPUTATION_NOT_STARTED |
| 0xC002 | Severe | PUKCL_UNKNOWN_SERVICE |
| 0xC003 | Severe | PUKCL_UNEXPLOITABLE_OPTIONS |
| 0xC004 | Severe | PUKCL_HARDWARE_ISSUE |
| 0xC005 | Severe | PUKCL_WRONG_HARDWARE |
| 0xC006 | Severe | PUKCL_LIBRARY_MALFORMED |
| 0xC007 | Severe | PUKCL_ERROR |
| 0xC008 | Severe | PUKCL_UNKNOWN_SUBSERVICE |
| 0xC101 | Severe | PUKCL_DIVISION_BY_ZERO |
| 0xC102 | Severe | PUKCL_MALFORMED_MODULUS |
| 0xC103 | Severe | PUKCL_FAULT_DETECTED |
| 0xC104 | Severe | PUKCL_MALFORMED_KEY |

Please note the following rules about return codes:

- A status value indicating a severe error, means that an expected operation has not been executed or has been corrupted. Therefore, the result of such an operation should never be used.
- A status value indicating a warning should be looked at precisely, as the expected correctness of the result cannot be guaranteed.
- A status value indicating an information always means that the result is correct with no possible misinterpretation of the values.
- A status value zero indicates that there is no error or no severity.

In the following sections, for each service, the constraints on the parameters placement are detailed. For reduced code size and higher execution speed, tests are processed on these constraints. It is important that PUKCL users take these placement constraints into consideration at the development and test stages to ensure the correct functioning of the library.

### 43.3.4. Basic Arithmetic and Cryptographic Services

#### 43.3.4.1. SelfTest

##### 43.3.4.1.1. Purpose

This service is used to initialize the PUKCL. It resets the PUKCC, clears the Crypto RAM, and returns the library and PUKCC version numbers.

It must be called before using any other services in the library and the user must verify the return status at the end of the service execution.

##### 43.3.4.1.2. How to Use the Service

##### 43.3.4.1.3. Description

This service processes internal tests and returns information and status codes as described in Status Returned Values. The service name for this operation is `SelfTest`.

##### 43.3.4.1.4. Parameters Definition

It is possible to directly address this service through the `PUKCL_SelfTest()` macro.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-5.** SelfTest Service Parameters

| Parameter | Type | Dir. | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u4Version | u4 | O | – | – | – | PUKCL version |
| u4PUKCCVersion | u4 | O | – | – | – | PUKCC Version |
| u4CheckNum1 | u4 | O | – | – | – | Test result value 1 |
| u4CheckNum2 | u4 | O | – | – | – | Test result value 2 |
| u1Step | u1 | O | – | – | – | Latest correctly executed step |

#### 43.3.4.1.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library
vPUKCL_Process(SelfTest,pvPUKCLParam);

if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The Library version is available
            // in PUKCL_SelfTest(u4Version)
            // The PUKCL version is available
            // in PUKCL_SelfTest(u4PUKCCVersion)
            }
```

#### 43.3.4.1.6. Returned Values

The expected u4Version value depends on the version of PUKCL being used, and the u4PUKCCVersion value depends on the version of PUKCC being used.

The expected u4CheckNum1 value is 0x6e70ddd2 and the expected one for u4CheckNum2 is 0x25c8d64f. The expected final u1Step value is 3.

#### 43.3.4.1.7. Status Returned Values

**Table 43-6.** SelfTest Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly. |
| PUKCL_ERROR | Severe | An issue has been encountered. |

### 43.3.4.2. Clear Flags

#### 43.3.4.2.1. Purpose

This service can be used to clear parameter structure flags.

#### 43.3.4.2.2. How to Use the Service

#### 43.3.4.2.3. Description

This service clears CarryOut, CarryIn, Zero and Violation flags in the Specific bit field. The Gf2n flag is untouched.

The service name for this operation is `ClearFlags`.

#### 43.3.4.2.4. Parameters Definition

It is possible to directly address this service through the `PUKCL_ClearFlags()` macro.

**Table 43-7.** Clear Flags Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| Specific/CarryOut | Bit | O | – | – | – | Cleared |
| Specific/CarryIn | Bit | O | – | – | – | Cleared |
| Specific/Zero | Bit | O | – | – | – | Cleared |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-7.** Clear Flags Service Parameters (continued)

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| Specific/Violation | Bit | O | – | – | – | Cleared |

#### 43.3.4.2.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ClearFlags,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // Success
            }
else // Manage the error
```

#### 43.3.4.2.6. Status Returned Values

**Table 43-8.** ClearFlags Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly. |

### 43.3.4.3. Swap

#### 43.3.4.3.1. Purpose

This service performs swapping of two buffers.

#### 43.3.4.3.2. How to Use the Service

#### 43.3.4.3.3. Description

This service swaps two buffers, X and Y, of the same size in memory.

The service name for this operation is `Swap`.

#### 43.3.4.3.4. Parameters Definition

This service can easily be accessed through the use of the `PUKCL_Swap()` macro.

**Table 43-9.** Swap Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1XBase | nu1 | I | Crypto RAM | u2Length | Base of the number X | Base of X filled with Y |
| nu1YBase | nu1 | I | Crypto RAM | u2Length | Base of the number Y | Base of Y filled with X |
| u2XLength | u2 | I | – | – | Length of X and Y | Length of X and Y |

#### 43.3.4.3.5. Code Example

```
_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// Initialize parameters
PUKCL_Swap(nu1XBase) = <Base of the X number>;
PUKCL_Swap(nu1YBase) = <Base of the Y number>;
PUKCL_Swap(u2XLength) = <Length of the numbers>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Swap,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.4.3.6. Constraints

The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase or nu1YBase are not aligned on 32-bit boundaries
- u2XLength is either <4, > 0xffc, or not a 32-bit length
- {nu1XBase, u2XLength} or {nu1YBase, u2XLength} do not entirely lie in PUKCCRAM
- {nu1XBase, u2XLength} overlaps {nu1YBase,u2YLength}

#### 43.3.4.3.7. Status Returned Values

**Table 43-10.** Swap Service Return Codes

| Returned status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |

### 43.3.4.4. Fill

#### 43.3.4.4.1. Purpose

This service performs a memory fill operation, with a given 32-bit constant.

#### 43.3.4.4.2. How to Use the Service

#### 43.3.4.4.3. Description

This service fills a Crypto RAM space with a provided 32-bit constant: Fill (R, FillValue)

The service name for this operation is `Fill`.

#### 43.3.4.4.4. Parameters Definition

This service can easily be accessed through the use of the `PUKCL_Fill()` macro.

**Table 43-11.** Fill Service Parameters

| Parameter | Type | Direction. | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1RBase | nu1 | I | Crypto RAM | u2RLength | Base of R | Base of R value filled repetitively with u4FillValue |
| u2RLength | u2 | I | Crypto RAM | – | Length of R | Length of R |
| u4FillValue | u4 | I | – | – | Filling value | Filling value |

#### 43.3.4.4.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// Initialize parameters
PUKCL_Fill(nu1RBase) = <Base of the R number>;
PUKCL_Fill(u2RLength) = <Length of the R number>;
PUKCL_Fill(u4FillValue) = <32-bits value to fill with>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Fill,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.4.4.6. Constraints

The following conditions must be avoided to ensure that the service works correctly:

- nu1RBase are not aligned on 32-bit boundaries
- u2RLength is either: <4, >0xffc or not a 32-bit length


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- {nu1RBase, u2RLength} do not entirely lie in Crypto RAM

#### 43.3.4.4.7. Status Returned Values

**Table 43-12.** Fill Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly. |

### 43.3.4.5. Fast Copy/Clear

#### 43.3.4.5.1. Purpose

This service performs a copy from a memory area to another or a memory area clear.

#### 43.3.4.5.2. How to Use the Service

#### 43.3.4.5.3. Description

This service copies a number X into another number R, padding with zero on the MSB side up to the length specified for R.

$R = X$

If the lengths of R and X are equal, a complete fast copy is processed.

If the length of R is strictly greater than the length of X, X is first copied in the Low Significant Bytes side of R, and R is padded with zeros on the Most Significant Bytes side.

If the pointer on the X area equals zero, R is filled with zeros. This operation can also be made by using the Fill service (see Fill).

The service name for this operation is `FastCopy`.

**Important:** The length of R must be greater or equal to the length of X.

#### 43.3.4.5.4. Parameters Definition

This service can easily be accessed through the use of the `PUKCL_FastCopy()` macro.

**Table 43-13.** FastCopy Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1XBase | nu1 | I | Crypto RAM | u2XLength | Base of X | Base of X number untouched |
| nu1RBase | nu1 | I | Crypto RAM | u2RLength | Base of R | Base of R filled with X |
| u2RLength | u2 | I | – | – | Length of R | Length of R |
| u2XLength | u2 | I | – | – | Length of X | Length of X |

#### 43.3.4.5.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// Initialize parameters
PUKCL_FastCopy(nu1XBase) = <Base of the X number>;
PUKCL_FastCopy(nu1RBase) = <Base of the R number>;
PUKCL_FastCopy(u2XLength) = <Length of the X number>;
PUKCL_FastCopy(u2RLength) = <Length of the R number>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(FastCopy,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            ...
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
                }
else // Manage the error
```

#### 43.3.4.5.6. Constraints

The parameter placements that are not allowed are are as follows.

If nu1XBase equals zero, no checks are made on nu1XBase (fixed) and u2XLength (unused).

The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase or nu1RBase are not aligned on 32-bit boundaries
- u2XLength or u2RLength is either: <4, >0xffc or not a 32-bit length or u2XLength >u2RLength
- {nu1XBase, u2XLength} or {nu1RBase, u2RLength} do not entirely lie in Crypto RAM
- {nu1XBase, u2XLength} overlaps {nu1RBase,u2RLength}

#### 43.3.4.5.7. Status Returned Values

**Table 43-14.** FastCopy Service Return Codes

| Returned status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |

### 43.3.4.6. Conditional Copy/Clear

#### 43.3.4.6.1. Purpose

This service conditionally performs a copy from a memory area to another or a memory area clear.

#### 43.3.4.6.2. How to Use the Service

#### 43.3.4.6.3. Description

This service copies a number X into another number R, padding with zero on the MSB side up to the length specified for R. This copy operation is performed under the conditions specified in the options.

If the condition is verified, R = X.

The copy or clear action is made under condition.

The four possible options for the condition are described in the following table. Two of the conditions check the Specific.CarryIn bit (see Accessing Different Library Services).

The processing is done as follows:

- If the condition is not verified, nothing is processed.
- If the condition is verified the copy or clear follows the rules:
  - If the lengths of R and X are equal, a complete fast copy is processed
  - If the length of R is strictly greater than the length of X, X is first copied in the Low Significant Bytes side of R, and R is padded with zeros on the Most Significant Bytes side.
  - If the pointer on the X area equals zero, R is filled with zeros.

The service name for this operation is `CondCopy`.

**Important:** If the condition is verified, the length of R must be greater or equal to the length of X.

#### 43.3.4.6.4. Parameters Definition

This service can easily be accessed through the use of the `PUKCL_CondCopy()` and `PUKCL()` macros.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-15.** CondCopy Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Option for condition (see the following table) | Option for condition (see the following table) |
| Specific/CarryIn | Bit | I | – | – | Bit CarryIn | Bit CarryIn |
| nu1XBase | nu1 | I | Crypto RAM | u2XLength | Base of X | Base of X number untouched |
| nu1RBase | nu1 | I | Crypto RAM | u2RLength | Base of R | Base of R filled with X if condition holds |
| u2RLength | u2 | I | – | – | Length of R | Length of R |
| u2XLength | u2 | I | – | – | Length of X | Length of X |

### 43.3.4.6.5. Available Options

The option for the condition is set by the u2Options input parameter that must take one of the values listed in the following table.

**Table 43-16.** CondCopy Service Options

| Option | Purpose | Needed parameters |
|---|---|---|
| PUKCL_CONDCOPY_ALWAYS | Always perform the copy | nu1XBase,u2XLength,nu1RBase, u2RLength |
| PUKCL_CONDCOPY_NEVER | Never perform the copy | None |
| PUKCL_CONDCOPY_IF_CARRY | Perform the copy if CarryIn is 1 | Specific/CarryIn nu1XBase,u2XLength,nu1RBase, u2RLength |
| PUKCL_CONDCOPY_IF_NOT_CARRY | Perform the copy if CarryIn is zero | Specific/CarryIn nu1XBase,u2XLength,nu1RBase, u2RLength |

### 43.3.4.6.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// CarryIn shall be beforehand filled (with zero or one) PUKCL(Specific).CarryIn = ...;

// Condition Option PUKCL(u2Options) = ...;

// Initialize parameters
PUKCL_CondCopy(nu1XBase) = <Base of the X number>;
PUKCL_CondCopy(nu1RBase) = <Base of the R number>;
PUKCL_CondCopy(u2XLength) = <Length of the X number>;
PUKCL_CondCopy(u2RLength) = <Length of the R number>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(CondCopy,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

### 43.3.4.6.7. Constraints

The parameters placement that are not allowed are listed below.

If the conditional option and the CarryIn do not lead to execute the copy, no checks are made on the constraints to be respected.

If nu1XBase equals zero, no checks are made on nu1XBase (fixed) and u2XLength (unused).

The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase or nu1RBase are not aligned on 32-bit boundaries
- u2XLength or u2RLength is either: <4, >0xffc or not a 32-bit length or u2XLength >u2RLength
- {nu1XBase, u2XLength} or {nu1RBase, u2RLength} do not entirely lie in Crypto RAM


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- {nu1XBase, u2XLength} overlaps {nu1RBase,u2RLength}

#### 43.3.4.6.8. Status Returned Values

**Table 43-17.** CondCopy Service Return Codes

| Returned status | Importance | Meaning |
|---|---|---|
| PUKCL_WRONG_SERVICE | Severe | An inconsistency has been detected between the called service and the provided service number. |
| PUKCL_OK | – | Service functioned correctly |

### 43.3.4.7. Small Multiply, Add, Subtract, Exclusive OR

#### 43.3.4.7.1. Purpose

This purpose of this service is to multiply a large number X by a single-word number, MulValue, and perform an optional accumulation/subtract with a large number Z, returning the result R.

The following options are available:

- Work in the GF($2^n$) or in the standard GF(p) arithmetic integer field
- Add of a supplemental CarryOperand
- Overlap of the operands is possible, taking into account some constraints
- Modulo-reduction of the computation result (see Modular Reduction)

In addition to a multiply, possible uses of this service can include:

- Copy a block of data from one place to another (if u4MulValue is 1). This operation can alternatively be made by using the Fast Copy service (see Fast Copy/Clear).
- Adding/Subtracting two numbers (if u4MulValue is1)
- Xoring two blocks of data (if u4MulValue is 1 and the selected mathematical field is GF($2^n$))

#### 43.3.4.7.2. How to Use the Service

#### 43.3.4.7.3. Description

This service processes the following operation (if not computing a modular reduction of the result):

$R = [Z] \pm (MulValue \times X + CarryOperand)$

Or (if computing a modular reduction of the result):

$R = ([Z] \pm (MulValue \times X + CarryOperand)) mod\ N$

The service name for this operation is `Smult`.

The result of the Small Multiply Operation is stored on u2RLength bytes, so the choice of this length compared to u2XLength may lead to:

- A truncation if the result is too big to be stored on u2RLengthbytes.
- A padding on the MSB side if the result does not take all the u2RLengthbytes.
  However, in all cases this rule must be followed:

> **Important:** The length of R must be greater than or equal to the length of X.

In these computations, the following parameters need to be provided:

- R the result (pointed by{nu1RBase,u2Rlength})
- X one input number or GF($2^n$) polynomial (pointed by{nu1XBase,u2XLength})
- Z one optional input number or GF($2^n$) polynomial (pointed by{nu1ZBase,u2Rlength}).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- MulValue one input number or GF($2^n$)polynomial on one word (provided in u4MulValue)
- CarryOperand (provided through the CarryOptions and Carry values).

**Important:** Even if neither accumulation nor subtraction is specified, the nu1ZBase must always be filled and point to a Crypto RAM space. It this case, nu1ZBase can point to the same space as the nu1RBase.

If using the modular reduction option, the Multiply operation is followed by a reduction (see Modular Reduction) and the following parameters must be additionally provided:

- N—the modulus (pointed by {nu1ModBase,u2Modlength +4})
- Cns—the reduction constant
  - In case of Big reduction, Cns is pointed by {nu1CnsBase,64bytes}.
  - In case of Fast or Normalized reduction, Cns is pointed by {nu1CnsBase,u2ModLength +8}

**Important:**
The result buffer R must first be padded with zero bytes until its length is sufficient to perform the reduction (2*u2ModLength + 8) to be used by the Modular Reduction service as an input parameter.

The result of the reduction is written in the area X pointed by {nu1XBase, u2ModLength + 4}.

- For example, if relevant u2ModLength is 0x80 bytes and u2XLength is 0x80 too, the length of the Rspace may be 2*(u2ModLength + 4) = 0x108 bytes.
  In case of fast or normalized reduction, the length of the result may be u2ModLength + 4 so 0x84 bytes. Therefore, the zone X may lengths 0x84 bytes (at least). The multiplication of X by 1 word provide a result in the zone R which MSB bytes will be padded with zero bytes.
  In that example, the length of the zone R will be 2*u2ModLength + 8 = 0x108 bytes.

### 43.3.4.7.4. Parameters Definition

**Table 43-18.** Smult Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |
| Specific/Gf2n CarryIn | Bits | I | – | – | GF($2^n$) Bit and Carry In | – |
| Specific/CarryOut Zero Violation | Bits | I | – | – | – | Carry Out, Zero Bit and Violation Bit filled according to the result |
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of N | Base of N untouched |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns untouched |
| u2ModLength | u2 | I | – | – | Length of N | Length of N |
| nu1XBase | nu1 | I | Crypto RAM | u2XLength or u2ModLength + 4 (see **Note 1**) | Base of X | Base of X ( see **Note 2**) |
| u2XLength | u2 | I | – | – | Length of X | Length of X |
| nu1ZBase | nu1 | I | Crypto RAM | u2RLength | Base of Z | Base of Z untouched |
| nu1RBase | nu1 | I | Crypto RAM | u2RLength | Base of R | Base of R (see **Note 3**) |
| u2RLength | u2 | I | – | – | Length of R | Length of R |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-18.** Smult Service Parameters (continued)

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u4MulValue | u4 | I | – | – | Value of MulValue | Value of MulValue untouched |

**Notes:**

1. If a reduction option is specified, the area X will be, if necessary, extended to u2ModLength + 4 bytes.
2. If Smult is without reduction, X is untouched. If Smult is with reduction, X is filled with the final result.
3. If Smult is without reduction, R is filled with the final result. If Smult is with reduction, R is corrupted.

### 43.3.4.7.5. Available Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Small Multiplication operation option described in Table 43-19
- the mandatory CarryOperand option described in Table 43-20 and Table 43-21
- the facultative Modular Reduction option (see Modular Reduction). If the Modular Reduction is not requested, this option is absent.

The u2Options number is calculated by an “Inclusive OR” of the options. Some examples in C language are:

- Operation: Small Multiply only without carry and without Modular Reduction
  ```
  PUKCL(u2Options) = SET_MULTIPLIEROPTION(PUKCL_SMULT_ONLY) |
  SET_CARRYOPTION(CARRY_NONE);
  ```
- Operation: Small Multiply with addition with Specific/CarryIn addition and with Fast Modular Reduction
  ```
  PUKCL(u2Options) =SET_MULTIPLIEROPTION(PUKCL_SMULT_ADD) |
  SET_CARRYOPTION(ADD_CARRY) | PUKCL_REDMOD_REDUCTION |
  PUKCL_REDMOD_USING_FASTRED;
  ```

The following table lists all of the necessary parameters for the Small Multiply option. When the Addition or Subtraction option is not chosen, it is not necessary to fill in the nu1ZBase parameter.

**Table 43-19.** Smult Service Operation Options

| Option | Purpose | Required Parameters |
|---|---|---|
| SET_MULTIPLIEROPTION(PUKCL_SMULT_ ONLY) | Perform R = MulValue*X + CarryOperand | nu1RBase, u2RLength, nu1XBase, u2XLength, u4MulValue |
| SET_MULTIPLIEROPTION(PUKCL_SMULT_ ADD) | Perform R = Z + MulValue*X + CarryOperand | nu1RBase, u2RLength, nu1ZBase, nu1XBase, u2XLength,u4MulValue |
| SET_MULTIPLIEROPTION(PUKCL_SMULT_ SUB) | Perform R = Z - (MulValue*X + CarryOperand) | nu1RBase, u2RLength, nu1ZBase, nu1XBase, u2XLength,u4MulValue |

### 43.3.4.7.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// Gf2n and CarryIn shall be beforehand filled (with zero or one)
PUKCL(Specific).Gf2n = ...;
PUKCL(Specific).CarryIn = ...; PUKCL(u2Options) =...;

// Depending on the option specified, not all fields should be filled
PUKCL_Smult(nu1XBase) = <Base of the X number>;
PUKCL_Smult(u2XLength) = <Length of the X number>;
PUKCL_Smult(nu1RBase) = <Base of the R number>;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL_Smult(u2RLength) = <Length of the R number>;
PUKCL_Smult(nu1ZBase) = <Base of the Z number>;
PUKCL_Smult(u4MulValue) = <Value to be multiplied with>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Smult,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The Small multiplication has been executed correctly
            ...
            }
else // Manage the error
```

**Note:**
The length of R must be greater or equal to the length of X. Additional options are available through the use of a modular reduction to be executed at the end of this operation. Some important considerations have to be taken into account concerning the length of resulting operands to get a mathematically correct result.

The output of this operation is not obviously compatible with the modular reduction, as it may be either smaller or bigger. In the case (most of the time) where the result (pointed by nu1RBase) is smaller in size than twice the modulus plus one word, it is mandatory to add padding bytes to zero. Otherwise, the reduced value will be taken considering the high order words (potentially uninitialized) as part of the number, thus resulting in a mathematically correct but unexpected result.

In the case that the result is bigger than twice the modulus plus one word, the modular reduction feature has to be executed as a separate operation, using an Euclidean division.

#### 43.3.4.7.7. Constraints

For the case of a small multiplication with an option indicating either subtraction or accumulation, the following conditions must be avoided to ensure the service works correctly:

- nu1XBase, nu1RBase or nu1ZBase are not aligned on 32-bit boundaries
- {nu1XBase, u2XLength}, {nu1ZLength, u2RLength} or {nu1RBase, u2RLength} do not entirely lie in Crypto RAM
- u2XLength or u2RLength is either: < 4, > 0xffc or not a 32-bit length or u2XLength >u2RLength
- {nu1RBase, u2RLength} overlaps {nu1XBase, u2XLength} or nu1R < nu1Z and {nu1RBase,u2RLength} overlaps {nu1ZBase, u2RLength}

If the nu1R value is greater or equals to the nu1Z one, the overlapping between R and Z is allowed.

If a modular reduction is specified, the relevant parameters must be defined according to the chosen reduction and follow the description in Modular Reduction. Additional constraints to be respected and error codes are described in this section and in Table 43-22.

**Multiplication with Accumulation or Subtraction**
When the options bits specify that either an Accumulation or a Subtraction should be performed, this service performs the following operation:

$R = (Z \pm (MulValue \times X + CarryOperand)) mod\ B^{RLength}$

**Table 43-20.** Smult Service (with Accumulate/Subtract From) Carry Settings

| Carry Options | CarryOperand | Resulting Operation |
|---|---|---|
| SET_CARRYOPTION(ADD_CARRY) | CarryIn | R = Z ± (MulValue*X + CarryIn) |
| SET_CARRYOPTION(SUB_CARRY) | - CarryIn | R = Z ± (MulValue*X - CarryIn) |
| SET_CARRYOPTION(ADD_1_PLUS_CARRY) | 1 + CarryIn | R = Z ± (MulValue*X + 1 + CarryIn) |
| SET_CARRYOPTION(ADD_1_MINUS_CARRY) | 1 - CarryIn | R = Z ± (MulValue*X + 1 - CarryIn) |
| SET_CARRYOPTION(CARRY_NONE) | 0 | R = Z ± (MulValue*X) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-20.** Smult Service (with Accumulate/Subtract From) Carry Settings (continued)

| Carry Options | CarryOperand | Resulting Operation |
|---|---|---|
| SET_CARRYOPTION(ADD_1) | 1 | R = Z ± (MulValue*X + 1) |
| SET_CARRYOPTION(SUB_1) | - 1 | R = Z ± (MulValue*X - 1) |
| SET_CARRYOPTION(ADD_2) | 2 | R = Z ± (MulValue*X + 2) |

### Multiplication without Accumulation or Subtraction

When the case the options bits specify that neither an Accumulation nor a Subtraction should be performed, this service performs the following operation:

$R = (MulValue \times X + CarryOperand) mod\ B^{RLength}$

**Table 43-21.** Smult Service Carry Settings

| Carry Options | CarryOperand | Resulting Operation |
|---|---|---|
| SET_CARRYOPTION(ADD_CARRY) | CarryIn | R = MulValue*X + CarryIn |
| SET_CARRYOPTION(SUB_CARRY) | - CarryIn | R = MulValue*X - CarryIn |
| SET_CARRYOPTION(ADD_1_PLUS_CARRY) | 1 + CarryIn | R = MulValue*X + 1 + CarryIn |
| SET_CARRYOPTION(ADD_1_MINUS_CARRY) | 1 - CarryIn | R = MulValue*X + 1 - CarryIn |
| SET_CARRYOPTION(CARRY_NONE) | 0 | R = MulValue*X |
| SET_CARRYOPTION(ADD_1) | 1 | R = MulValue*X + 1 |
| SET_CARRYOPTION(SUB_1) | -1 | R = MulValue*X - 1 |
| SET_CARRYOPTION(ADD_2) | 2 | R = MulValue*X + 2 |

## 43.3.4.7.8. Status Returned Values

**Table 43-22.** Smult Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |

## 43.3.4.8. Compare

### 43.3.4.8.1. Purpose

The purpose of this service is to compare two numbers in classical arithmetic GF(p).

**Important:** This service works only with integers.

### 43.3.4.8.2. How to Use the Service

### 43.3.4.8.3. Description

This service accepts two numbers in classical arithmetic in input and performs a comparison, virtually subtracting (X + CarryIn) from Y:

CompareGetFlags (Y - (X + CarryIn))

The numbers X and Y are untouched but the resulting flags CarryOut and the Zero Bit are filled. If the lengths of Y and X are equal, a comparison is processed.

If the length of Y is strictly greater than the length of X, X is first virtually padded with zeros on the Most Significant Bytes side, then a comparison is processed.

**Note:** The length of Y must be greater or equal to the length of X.

In this computation, the following data need to be provided:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- X (pointed by{nu1XBase,u2XLength})
- Y (pointed by{nu1YBase,u2YLength})

The service name for this operation is `Comp`.

#### 43.3.4.8.4. Parameters Definition

**Table 43-23.** Comp Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| Specific/Gf2n CarryIn | Bits | I | – | – | GF(2n) Bit and Carry In | – |
| Specific/CarryOut Zero Violation | Bits | I | – | – | – | Carry Out, Zero Bit and Violation Bit filled according to the result |
| nu1XBase | nu1 | I | Crypto RAM | u2XLength | Base of X | Base of X |
| u2XLength | u2 | I | – | – | Length of X | Length of X |
| nu1YBase | nu1 | I | Crypto RAM | u2YLength | Base of Y | Base of Y |
| u2YLength | u2 | I | – | – | Length of Y | Length of Y |

#### 43.3.4.8.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// CarryIn shall be beforehand filled (with zero or one) PUKCL(Specific).CarryIn = ...;

// Initializing parameters
PUKCL_Comp(nu1XBase) = <Base of the ram location of X>;
PUKCL_Comp(u2XLength) = <Length of X>;
PUKCL_Comp(nu1YBase) = <Base of the ram location of Y>;
PUKCL_Comp(u2YLength) = <Length of Y>;

// vPUKCL_Process() is a macro command,
// and then calls the library...
vPUKCL_Process(Comp,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The COMPARE has been executed correctly
            // CarryOut, Zero ... are available
            ... = PUKCL(Specific).CarryOut;
            ... = PUKCL(Specific).Zero;
            }
else // Manage the error
```

#### 43.3.4.8.6. Constraints

The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase or nu1YBase are not aligned on 32-bit boundaries
- {nu1XBase, u2XLength} or {nu1YLength, u2YLength} are not in Crypto RAM
- u2XLength or u2YLength is either: < 4, > 0xffc or not a 32-bit length or u2XLength >u2YLength

#### 43.3.4.8.7. Status Returned Values

**Table 43-24.** Comp Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |

### 43.3.4.9. Full Multiply

#### 43.3.4.9.1. Purpose

The purpose of this service is to multiply two large numbers, X and Y, and optionally accumulate/subtract from a third large number, Z, returning the result, R.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The available options are as follows:

- Work in the GF($2^n$) field or in the standard arithmetic field
- Add of a supplemental CarryOperand
- Overlap of the operands is possible, taking into account some constraints
- Modular Reduction of the computation result (see Modular Reduction)

#### 43.3.4.9.2. How to Use the Service

#### 43.3.4.9.3. Description

This service provides the following (if not computing a modular reduction of the result):

$R = [Z] \pm (X \times Y + CarryOperand)$

Or (if computing a modular reduction of the result):

$R = ([Z] \pm (X \times Y + CarryOperand)) mod\ N$

The service name for this operation is `Fmult`.

In these computations, the following data has to be provided:

- R the result (pointed by {nu1RBase,u2Xlength +u2YLength})
- X one input number or GF($2^n$) polynomial (pointed by{nu1XBase,u2XLength})
- Y one input number or GF($2^n$) polynomial (pointed by{nu1YBase,u2YLength})
- Z one optional input number or GF(2n) polynomial (pointed by {nu1ZBase,u2Xlength +u2YLength})
- CarryOperand (provided through the Carry Options and Carry values)

**Important:** Even if neither accumulation nor subtraction is specified, the nu1ZBase must always be filled and point to a Crypto RAM space. It this case, nu1ZBase can point to the same space as the nu1RBase.

If using the big modular reduction option, the Multiply operation is followed by a reduction (see Modular Reduction). In this case, the length of Cns is 64 bytes.

If using the modular reduction option, the Multiply operation is followed by a reduction (see Modular Reduction). In this case the following parameters must be additionally provided:

- N—the modulus (pointed by {nu1ModBase,u2Modlength +4})
- Cns—the reduction constant
  - In case of Big reduction, Cns is pointed by {nu1CnsBase,64bytes}.
  - In case of Fast or Normalized reduction, Cns is pointed by (pointed by {nu1CnsBase,u2ModLength+ 8})

**Note:**
The result buffer R must first be padded with zero bytes until its length is sufficient to perform the reduction (2*u2ModLength + 8) to be used by the Modular Reduction service as an input parameter.

The result of the reduction is written in the area X pointed by {nu1XBase, u2ModLength + 4}.

For example, if u2ModLength, u2XLength and u2YLength are 0x80 bytes, the length of the R space is 2*(u2ModLength + 4) = 0x108 bytes because of the constraints of modular reduction.

In case of Fast or Normalized Reduction, the length of the result is u2ModLength + 4 so 0x84 bytes. Thus, the zone X has a length of 0x84 bytes (at least). The multiplication of X by Y provides a result of


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

length 0x100 bytes in the zone R so the 8 MSB bytes must be previously padded with zero bytes (in offsets 0x100 to 0x107).

#### 43.3.4.9.4. Parameters Definition

**Table 43-25.** Fmult Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |
| Specific/Gf2n CarryIn | Bits | I | – | – | GF(2n) Bit and Carry In | – |
| Specific/CarryOut Zero Violation | Bits | I | – | – | – | Carry Out, Zero Bit and Violation Bit filled according to the result |
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of N | Base of N untouched |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 or 64 bytes | Base of Cns | Base of Cns untouched |
| u2ModLength | u2 | I | – | – | Length of N | Length of N |
| nu1XBase | nu1 | I | Crypto RAM | u2XLength or u2ModLength + 4 (see **Note 1**) | Base of X | Base of X (see **Note 2**) |
| u2XLength | u2 | I | – | – | Length of X | Length of X |
| nu1YBase | nu1 | I | Crypto RAM | u2YLength | Base of Y | Base of Y |
| u2YLength | u2 | I | – | – | Length of Y | Length of Y |
| nu1ZBase | nu1 | I | Crypto RAM | u2XLength + u2YLength | Base of Z | Base of Z untouched |
| nu1RBase | nu1 | I | Crypto RAM | u2XLength + u2YLength | Base of R | Base of R (see **Note 3**) |

**Notes:**

1. In case of a reduction option is specified, if necessary, the area X will be extended to u2ModLength + 4 bytes.
2. If FMult is without reduction, X is untouched. If FMult is with reduction, X is filled with the final result.
3. If FMult is without reduction, R is filled with the final result. If FMult is with reduction, R is corrupted.

#### 43.3.4.9.5. Available Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Full Multiplication operation option described in Table 43-26
- the mandatory CarryOperand option described in Table 43-27 and Table 43-28
- the facultative Modular Reduction option (see Modular Reduction). If the Modular Reduction is not requested, this option is absent.

The u2Options number is calculated by an Inclusive OR of the options.

Some Examples in C language are:

- Operation: Full Multiply only without carry and without Modular Reduction
```
PUKCL(u2Options) = SET_MULTIPLIEROPTION(PUKCL_FMULT_ONLY) |
SET_CARRYOPTION(CARRY_NONE);
```
- Operation: Full Multiply with addition with Specific/CarryIn addition and with Fast Modular Reduction
```
PUKCL(u2Options) = SET_MULTIPLIEROPTION(PUKCL_FMULT_ADD) |
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
SET_CARRYOPTION(ADD_CARRY) |
PUKCL_REDMOD_REDUCTION |
PUKCL_REDMOD_USING_FASTRED;
```

The following table shows all of the necessary parameters for the Full Multiply option. When the Addition or Subtraction option is not chosen, it is not necessary to fill in the nu1ZBase parameter.

**Table 43-26.** Fmult Service Options

| Option | Purpose | Required Parameters |
|---|---|---|
| SET_MULTIPLIEROPTION(PUKCL_FMUL_ONLY) | Perform R = X*Y + CarryOperand | nu1RBase, nu1YBase, u2YLength, nu1XBase, u2XLength |
| SET_MULTIPLIEROPTION(PUKCL_FMUL_ADD) | Perform R = Z + X*Y + CarryOperand | nu1RBase, nu1ZBase, nu1YBase, u2YLength, nu1XBase, u2XLength |
| SET_MULTIPLIEROPTION(PUKCL_FMUL_SUB) | Perform R = Z - (X*Y + CarryOperand) | nu1RBase, nu1ZBase, nu1YBase, u2YLength, nu1Xlength, u2XLength |

### 43.3.4.9.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


// Gf2n and CarryIn shall be beforehand filled (with zero or one)
PUKCL(Specific).Gf2n = ...;
PUKCL(Specific).CarryIn = ...;

PUKCL(u2Option) =...;
// Depending on the option specified, not all fields should be filled
PUKCL_Fmult(nu1XBase) = <Base of the ram location of X>;
PUKCL_Fmult(u2XLength) = <Length of X>;
PUKCL_Fmult(nu1YBase) = <Base of the ram location of Y>;
PUKCL_Fmult(u2YLength) = <Length of Y>;
PUKCL_Fmult(nu1ZBase) = <Base of the ram location of Z>;
PUKCL_Fmult(nu1RBase) = <Base of the ram location of R>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Fmult,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The Full multiply has been executed correctly
            ...
            }
else // Manage the error
```

### 43.3.4.9.7. Important Considerations for Modular Reduction of a Fmult Computation Result

**Note:**
Additional options are available through the use of a modular reduction to be executed at the end of this operation. Some important considerations have to be taken into account concerning the length of resulting operands to get a mathematically correct result.

The output of this operation is not always compatible with the modular reduction as it may be either smaller or bigger. In the case (most of the time) the result (pointed by nu1RBase) is smaller in size than “twice the modulus plus one word” by one word, a padding word must be added to zero. Otherwise, the reduced value will be taken considering the high order words (potentially uninitialized) as part of the number, thus resulting in getting a mathematically correct but unexpected result.

In the case that the result is bigger than twice the modulus plus one word, the modular reduction feature has to be executed as a separate operation, using an Euclidean division.

### 43.3.4.9.8. Constraints

The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase, nu1YBase, nu1RBase or nu1ZBase are not aligned on 32-bit boundaries


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- {nu1XBase, u2XLength}, {nu1YLength, u2YLength}, {nu1ZBase, u2XLength+u2YLength} or{nu1RBase, u2XLength+u2YLength} are not in Crypto RAM
- u2XLength, u2YLength is either: < 4, > 0xffc or not a 32-bit length
- {nu1RBase, u2XLength+u2YLength} overlaps {nu1YBase, u2YLength} or{nu1RBase, u2XLength+u2YLength} overlaps {nu1XBase, u2XLength}
- {nu1RBase, u2XLength+u2YLength} overlaps {nu1ZBase, u2XLength+u2YLength} and nu1RBase> nu1ZBase

If a modular reduction is specified, the relevant parameters must be defined according to the chosen reduction and follow the description in Modular Reduction. Additional constraints to be respected and error codes are described in this section and in Table 43-49.

### Multiplication with Accumulation or Subtraction

In the case where the options bits specify that either an Accumulation or a subtraction should be performed, this service performs the following operation:

$R = (Z \pm (X \times Y + CarryOperand)) mod\ B^{XLength + YLength}$

**Table 43-27.** Fmult Service (with Accumulate/Subtract From) Carry Settings

| Option AND CARRYOPTIONS | CarryOperand | Resulting Operation |
|---|---|---|
| SET_CARRYOPTION(ADD_CARRY) | CarryIn | R = Z ± (X*Y + CarryIn) |
| SET_CARRYOPTION(SUB_CARRY) | - CarryIn | R = Z ± (X*Y - CarryIn) |
| SET_CARRYOPTION(ADD_1_PLUS_CARRY) | 1 + CarryIn | R = Z ± (X*Y + 1 + CarryIn) |
| SET_CARRYOPTION(ADD_1_MINUS_CARRY) | 1 - CarryIn | R = Z ± (X*Y + 1 - CarryIn) |
| SET_CARRYOPTION(CARRY_NONE) | 0 | R = Z ± (X*Y) |
| SET_CARRYOPTION(ADD_1) | 1 | R = Z ± (X*Y + 1) |
| SET_CARRYOPTION(SUB_1) | - 1 | R = Z ± (X*Y - 1) |
| SET_CARRYOPTION(ADD_2) | 2 | R = Z ± (X*Y + 2) |

### Multiplication without Accumulation or Subtraction

In the case the options bits specify that either an Accumulation or a subtraction should be performed, this service performs the following operation:

$R = (X \times Y + CarryOperand) mod\ B^{XLength + YLength}$

**Table 43-28.** Fmult Service Carry Settings

| Option AND CARRYOPTIONS | CarryOperand | Resulting Operation |
|---|---|---|
| SET_CARRYOPTION(ADD_CARRY) | CarryIn | R = X*Y + CarryIn |
| SET_CARRYOPTION(SUB_CARRY) | - CarryIn | R = X*Y - CarryIn |
| SET_CARRYOPTION(ADD_1_PLUS_CARRY) | 1 + CarryIn | R = X*Y + 1 + CarryIn |
| SET_CARRYOPTION(ADD_1_MINUS_CARRY) | 1 - CarryIn | R = X*Y + 1 - CarryIn |
| SET_CARRYOPTION(CARRY_NONE) | 0 | R = X*Y |
| SET_CARRYOPTION(ADD_1) | 1 | R = X*Y + 1 |
| SET_CARRYOPTION(SUB_1) | - 1 | R = X*Y - 1 |
| SET_CARRYOPTION(ADD_2) | 2 | R = X*Y + 2 |

## 43.3.4.9.9. Status Returned Values

**Table 43-29.** Fmult Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 43.3.4.10. Square

#### 43.3.4.10.1. Purpose

The purpose of this service is to compute the square of a big number and optionally accumulate/subtract from a second big number.

Please note that this service uses an optimized implementation of the squaring. It also means that when the GF($2^n$) flag is set, the execution time will be smaller than when not set (in that case, the squaring execution time will still be smaller than for a standard multiplication).

The available options are as follows:

- Work in the GF($2^n$) or in the standard integer arithmetic field
- Add of a supplemental CarryOperand
- Overlapping of the operands is possible, taking into account some constraints
- Modular Reduction of the computation result

#### 43.3.4.10.2. How to Use the Service

#### 43.3.4.10.3. Description

This service provides the following (if not computing a modular reduction of the result):

$R = [Z] \pm (X^2 + CarryOperand)$

Or (if computing a modular reduction of the result):

$R = ([Z] \pm (X^2 + CarryOperand)) mod\ N$

The service name for this operation is `Square`.

In these computations, the following data has to be provided:

- R the result (pointed by {nu1RBase,2 *u2Xlength})
- X one input number or GF(2n) polynomial (pointed by{nu1XBase,u2XLength})
- Z one optional input number or GF(2n) polynomial (pointed by {nu1ZBase,2 *u2Xlength})
- CarryOperand (provided through the CarryOptions and Carry values)

**Important:** Even if neither accumulation nor subtraction is specified, the nu1ZBase must always be filled and point to a Crypto RAM space. It this case, nu1ZBase can point to the same space as the nu1RBase.

If using the big modular reduction option, the Multiply operation is followed by a reduction (see Modular Reduction). In this case, the length of Cns is 64 bytes.

If using the modular reduction option the Square operation is followed by a reduction (see Modular Reduction). In this case the following parameters must be additionally provided:

- N—the modulus (pointed by {nu1ModBase,u2Modlength +4}).
- Cns—the reduction constant (pointed by {nu1CnsBase,u2Modlength +8})
  - In case of big reduction option, the length of Cns is 64bytes.

**Note:**
The result buffer R must first be padded with zero bytes until its length is sufficient to perform the reduction (2*u2ModLength + 8) to be used by the Modular Reduction service as an input parameter.

The result of the reduction is written in the area X pointed by {nu1XBase, u2ModLength + 4}.

For example, if u2ModLength, u2XLength is 0x80 bytes, the length of the R space is 2*(u2ModLength + 4) = 0x108 bytes because of the constraints of modular reduction.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

In case of Fast or Normalized Reduction, the length of the result is u2ModLength + 4 so 0x84 bytes. Thus, the zoneX has a length of 0x84 bytes (at least). The square of X provides a result of length 0x100 bytes in the zone R so the 8 MSB bytes previously must be previously padded with zero bytes (in offsets 0x100 to 0x107).

### 43.3.4.10.4. Parameters Definition

**Table 43-30.** Square Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |
| Specific/Gf2n CarryIn | Bits | I | – | – | GF(2n) Bit and Carry In | – |
| Specific/CarryOut Zero Violation | Bits | I | – | – | – | Carry Out, Zero Bit and Violation Bit filled according to the result |
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of N | Base of N untouched |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 or 64 bytes | Base of Cns | Base of Cns untouched |
| u2ModLength | u2 | I | – | – | Length of N | Length of N |
| nu1XBase | nu1 | I | Crypto RAM | u2XLength or u2ModLength + 4 (see **Note 1**) | Base of X | Base of X ( see **Note 2**) |
| u2XLength | u2 | I | – | – | Length of X | Length of X |
| nu1ZBase | nu1 | I | Crypto RAM | 2 * u2XLength | Base of Z | Base of Z |
| nu1RBase | nu1 | I | Crypto RAM | 2 * u2XLength | Base of R | Base of R (see **Note 3**) |

**Notes:**

1. In case of a reduction option is specified, the area X will be, if necessary, extended to u2ModLength + 4 bytes.
2. If Square is without reduction, X is untouched. If Square is with reduction, X is filled with the final result.
3. If Square is without reduction, R is filled with the final result. If Square is with reduction, R is corrupted.

### 43.3.4.10.5. Available Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Square operation option described in Table 43-31
- the mandatory CarryOperand option described in Table 43-32 and Table 43-33
- the facultative Modular Reduction option (see Modular Reduction). If the Modular Reduction is not requested, this option is absent.

The u2Options number is calculated by an Inclusive OR of the options. Some Examples in C language are:

- Operation: Square only without carry and without Modular Reduction
```
PUKCL(u2Options) = SET_MULTIPLIEROPTION(PUKCL_SQUARE_ONLY) |
SET_CARRYOPTION(CARRY_NONE);
```
- Operation: Square with addition with Specific/CarryIn addition and with Fast Modular Reduction
```
PUKCL(u2Options) = SET_MULTIPLIEROPTION(PUKCL_SQUARE_ADD) |
SET_CARRYOPTION(ADD_CARRY) | PUKCL_REDMOD_REDUCTION |
PUKCL_REDMOD_USING_FASTRED;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The following table lists all of the necessary parameters for the Square option. When the Addition or Subtraction option is not chosen it is not necessary to fill in the nu1ZBase parameter.

**Table 43-31.** Square Service Options

| Option | Purpose | Required Parameters |
|---|---|---|
| SET_MULTIPLIEROPTION(PUKCL_ SQUARE_ONLY) | Perform R = $X^2$ + CarryOperand | nu1RBase, nu1ZBase, nu1XBase, u2XLength |
| SET_MULTIPLIEROPTION(PUKCL_ SQUARE_ADD) | Perform R = Z + $X^2$ + CarryOperand | nu1RBase, nu1ZBase, nu1XBase, u2XLength |
| SET_MULTIPLIEROPTION(PUKCL_ SQUARE_SUB) | Perform R = Z - ($X^2$ + CarryOperand) | nu1RBase, nu1ZBase, nu1Xlength, u2XLength |

### 43.3.4.10.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


// Gf2n and CarryIn shall be beforehand filled (with zero or one)
PUKCL(Specific).Gf2n = ...;
PUKCL(Specific).CarryIn = ...;

PUKCL(u2Option) =...;
// Depending on the option specified, not all fields should be filled
PUKCL_Fmult(nu1XBase) = <Base of the ram location of X>;
PUKCL_Fmult(u2XLength) = <Length of X>;
PUKCL_Fmult(nu1ZBase) = <Base of the ram location of Z>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Square,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The Squaring has been executed correctly
            ...
            }
else // Manage the error
```

### 43.3.4.10.7. Important Considerations for Modular Reduction of a Square Computation

**Note:**
Additional options are available through the use of a modular reduction to be executed at the end of this operation. Some important considerations have to be taken into account concerning the length of resulting operands to get a mathematically correct result.

The output of this operation is not obviously compatible with the modular reduction as it may be either smaller or bigger. In the case (most of the time) the result (pointed by nu1RBase) is smaller in size than “twice the modulus plus one word” by one word, a padding word must be added to zero. Otherwise, the reduced value will be taken considering the high order words (potentially uninitialized) as part of the number, thus resulting in getting a mathematically correct but unexpected result.

In the case that the result is greater than twice the modulus plus one word, the modular reduction feature has to be executed as a separate operation, using an Euclidean division.

### 43.3.4.10.8. Constraints

When the options only indicate a square, the constraints involving nu1ZBase are not checked. The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase, nu1RBase or nu1ZBase are not aligned on 32-bit boundaries
- {nu1XBase, u2XLength}, {nu1ZBase, 2*u2XLength} or {nu1RBase, 2*u2XLength} are not in Crypto RAM
- u2XLength is either: < 4, > 0xffc or not a 32-bit length


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- {nu1RBase, 2*u2XLength} overlaps {nu1XBase,u2XLength}
- {nu1RBase, 2*u2XLength} overlaps {nu1ZBase, 2*u2XLength} and nu1RBase >nu1ZBase

If a modular reduction is specified, the relevant parameters must be defined according to the chosen reduction and follow the description in Modular Reduction. Additional constraints to be respected and error codes are described in this section and in Table 43-49.

**Multiplication with Accumulation or Subtraction**

Where the options bits specify that either an Accumulation or a subtraction should be performed, this command performs the following operation:

$R = (Z \pm (X^2 + CarryOperand)) mod\ B^{2 * XLength}$

**Table 43-32.** Multiplication with Accumulation or Subtraction

| Option AND CARRYOPTIONS | CarryOperand | Resulting Operation |
|---|---|---|
| SET_CARRYOPTION(ADD_CARRY) | CarryIn | R = Z ± ($X^2$ + CarryIn) |
| SET_CARRYOPTION(SUB_CARRY) | - CarryIn | R = Z ± ($X^2$ - CarryIn) |
| SET_CARRYOPTION(ADD_1_PLUS_CARRY) | 1 + CarryIn | R = Z ± ($X^2$ + 1 + CarryIn) |
| SET_CARRYOPTION(ADD_1_MINUS_CARRY) | 1 - CarryIn | R = Z ± ($X^2$ + 1 - CarryIn) |
| SET_CARRYOPTION(CARRY_NONE) | 0 | R = Z ± ($X^2$) |
| SET_CARRYOPTION(ADD_1) | 1 | R = Z ± ($X^2$ + 1) |
| SET_CARRYOPTION(SUB_1) | - 1 | R = Z ± ($X^2$ - 1) |
| SET_CARRYOPTION(ADD_2) | 2 | R = Z ± ($X^2$ + 2) |

### 43.3.4.10.9. Multiplication without Accumulation or Subtraction

Where the options bits specify that either an accumulation or a subtraction should be performed, this command performs the following operation:

$R = (X^2 + CarryOperand) mod\ B^{2 * XLength}$

**Table 43-33.** Square Service Carry Settings

| Option AND CARRYOPTIONS | CarryOperand | Resulting Operation |
|---|---|---|
| SET_CARRYOPTION(ADD_CARRY) | CarryIn | R = $X^2$ + CarryIn |
| SET_CARRYOPTION(SUB_CARRY) | - CarryIn | R = $X^2$ - CarryIn |
| SET_CARRYOPTION(ADD_1_PLUS_CARRY) | 1 + CarryIn | R = $X^2$ + 1 + CarryIn |
| SET_CARRYOPTION(ADD_1_MINUS_CARRY) | 1 - CarryIn | R = $X^2$ + 1 - CarryIn |
| SET_CARRYOPTION(CARRY_NONE) | 0 | R = $X^2$ |
| SET_CARRYOPTION(ADD_1) | 1 | R = $X^2$ + 1 |
| SET_CARRYOPTION(SUB_1) | - 1 | R = $X^2$ - 1 |
| SET_CARRYOPTION(ADD_2) | 2 | R = $X^2$ + 2 |

### 43.3.4.10.10. Status Returned Values

**Table 43-34.** Square Service Return Codes

| Returned status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |

## 43.3.4.11. Integral (Euclidean) Division

### 43.3.4.11.1. Purpose

The purpose of this service is to compute the Euclidean Division of two multiple precision numbers in GF(p) or polynomial in GF($2^n$). The Numerator is divided by the Denominator giving the Quotient “Quo” and the Remainder “R”.

The following options are available:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Work in the GF($2^n$) field or in the standard integer arithmetic field GF(p)

#### 43.3.4.11.2. How to Use the Service

#### 43.3.4.11.3. Description

This service processes the calculus corresponding to:

$$Num = Mod \times Quo + R \quad with \quad 0 \leq R < Mod \quad and \quad Quo = \left[\frac{Num}{Mod}\right]$$

The Numerator is Num.

The Divisior (Modulus) is Mod.

The Quotient is Quo.

The Remainder is R.

The Inputs are, the Numerator Num, and the Denominator Mod. The service calculates the Quotient and the Remainder. The Remainder overwrites the Numerator and is copied to the R area.

If the parameter nu1QuoBase equals zero, the Quotient is not stored in memory.

If nu1QuoBase is different from zero, the Quotient length is (<Numerator Length> - <Denominator Length>) + 4 bytes.

In this computation, the following areas need to be provided:

- Num (pointed by {nu1NumBase,u2NumLength}) filled with the Numerator (with MSB word to zero).
- Mod (pointed by {nu1ModBase,u2ModLength}) filled with the Denominator.
- Workspace (pointed by {nu1CnsBase,64 or68}).
- Quo (pointed by {nu1QuoBase,u2NumLength - u2ModLength + 4}) to contain calculated Quotient.
  - When the quotient is not needed, the nu1QuoBase pointer can be provided as NULL. In that case, only the remainder will be provided as a result.
- R (pointed by {nu1RBase,u2ModLength}) to contain the calculated Remainder.

The service name for this operation is `Div`.

#### 43.3.4.11.4. Parameters Definition

**Table 43-35.** Div Service Parameters

| Parameter | Type | Dir. | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| Specific/Gf2n | Bit | I | – | – | GF(2n) Bit | – |
| nu1NumBase | nu1 | I | Crypto RAM | u2NumLength | Base of Num<br>Filled with the Numerator | Base of Num<br>Filled with the Remainder |
| u2NumLength | u2 | I | – | – | Length of the Numerator | Length of the Numerator |
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLengt | Base of the Divisor | Base of the Divisor untouched |
| u2ModLength | u2 | I | – | – | Length of the Divisor | Length of the Divisor |
| nu1QuoBase (see **Note 1**) | nu1 | I | Crypto RAM | u2NumLength - u2ModLength + 4 | Base of the Quotient | Base of the Quotient |
| nu1WorkSpace | nu1 | I | Crypto RAM | GF(p): 64<br>GF(2n): 68 | Base of the WorkSpace | Base of the WorkSpace corrupted |
| nu1RBase ( see **Note 2**) | nu1 | I | Crypto RAM | u2ModLength | Base of the Remainder | Base of the Remainder |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. If the quotient is not needed, set nu1QuoBase to zero and the quotient will not be written to memory. If the quotient is needed, set the nu1QuoBase to the beginning of an area of size (u2NumLength - u2ModLength + 4) to write the whole quotient.
2. The Remainder is present in the area {nu1NumBase, u2NumLength} at the end of the calculus. The nu1RBase parameter makes it possible to copy this result in the other area {nu1RBase, u2ModLength}, if this copy is not needed, set nu1RBase to the same value as nu1NumBase and the copy will not be done.

**Note:** The parameter Num must have its most significant 32-bit word cleared to zero. The length u2NumLength is the length of Num including this zero word.
One additional word is used on the LSB side of the Num parameter, this word is restored at the end of the calculus. As a consequence the parameter nu1NumBase must never been at the beginning of the Crypto RAM, i.e., ensure that nu1NumBase ≥ <Crypto RAM Base> + 4 bytes.

One additional word is used on the MSB side of the Num parameter, this word is not corrupted. As a consequence the Area {nu1NumBase, u2NumLength} must not be at the end of the Crypto RAM, i.e., en sure that nu1NumBase+u2NumLength ≤ <Crypto RAM End> - 4.

u2ModLength must be the true length of the Modulus, i.e., the MSB word of the area {nu1ModBase, u2ModLength} must be different from zero.

The minimum value for u2ModLength is 8 bytes, so the significant length of Num must be at least 8 bytes. To divide by a 32-bit value, the divider and numerator shall be multiplied by 232. The resulting remainder will have to be divided by $2^{32}$, the quotient will be exact.

### 43.3.4.11.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// Fill all the fields
// In that case, the quotient will be computed
// If it was not needed, set nu1QuoBase to NULL
PUKCL_Div(nu1NumBase) = <Base of the ram location of Num>;
PUKCL_Div(nu1ModBase) = <Base of the ram location of Mod>;
PUKCL_Div(nu1QuoBase) = <Base of the ram location of Quo>;
PUKCL_Div(nu1WorkSpace) = <Base of the workspace>;
PUKCL_Div(nu1RBase) = <Base of the ram location of R>;
PUKCL_Div(u2NumLength) = <Length of Num>;
PUKCL_Div(u2ModLength) = <Length of Mod>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Div,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The Division has been executed correctly
            ...
            }
else // Manage the error
```

### 43.3.4.11.6. Constraints

The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase, nu1RBase, nu1QuoBase, nu1WorkSpace or nu1NumBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength}, {nu1RBase, u2ModLength}, {nu1WorkSpace, 64} or{nu1NumBase, u2NumLength} are not in Crypto RAM
- u2ModLength, u2NumLength is either: < 4, > 0xffc or not a 32-bit length
- One or more overlaps exist between two of the areas: {nu1ModBase,u2ModLength},{nu1RBase, u2ModLength} {nu1NumBase, u2NumLength}(1) or {nu1WorkSpace,64}


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- If nu1QuoBase is different from zero and: {nu1QuoBase, u2NumLength - u2ModLength + 4} are not in Crypto RAM
- If nu1QuoBase is different from zero and one or more overlaps exist between two of the areas: {nu1QuoBase, u2NumLength - u2ModLength + 4}, {nu1ModBase, u2ModLength}, {nu1RBase, u2ModLength}, {nu1NumBase, u2NumLength} or {nu1WorkSpace, 64}

Overlaps between {nu1RBase, u2ModLength} and {nu1NumBase, u2NumLength} are forbidden, but the equality between nu1RBase and nu1NumBase is authorized

#### 43.3.4.11.7. Status Returned Values

**Table 43-36.** Div Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly. |
| PUKCL_DIVISION_BY_ZERO | Severe | The operation was not performed because the Denominator value is zero. |

### 43.3.4.12. GCD, Modular Inverse

#### 43.3.4.12.1. Purpose

The purpose of this command is to compute the Greatest Common Divisor (GCD) and the Modular Inverse. The algorithm used is the Extended Euclidean Algorithm for the GCD.

This command accepts as input two multiple precision numbers in GF(p) or two polynomials in GF($2^n$) X and Y and computes their GCD (D), if D equals one, the command also supplies the inverse of X modulo Y.

The available options are as follows:

- Work in the GF($2^n$) field or in the standard integer arithmetic field GF(p)

#### 43.3.4.12.2. How to Use the Service

#### 43.3.4.12.3. Description

This command calculates:

$D = GCD(X,Y)$.

and parameter A in the Bezout equation:

$A \times X + B \times Y = D$.

The first input, or input to inverse is X.

The second input, or modulus is Y.

The GCD is output in D.

The modular inverse if X and Y are co-primes is output A:

$A = X^{-1} mod(Y)$

The command calculates the GCD and the value A. The value A is the multiplicative inverse of X, only if X and Y are co-prime. As a supplemental result, Z is given back, being the quotient of Y divided by D only if D is different from zero:

$$Z = \left[\frac{Y}{D}\right]$$

At the end of the command: X is overwritten by D.

Y is cleared.

The value of A is calculated and stored.

The value of Z is calculated and stored if D is different from zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The service name for this operation is GCD.

In this computation, the following areas have to be provided:

- X (pointed by {nu1XBase,u2Length}) filled with X (with MSB word to zero)
- Y (pointed by {nu1YBase,u2Length}) filled with Y (with MSB word to zero)
- A (pointed by {nu1ABase,u2Length}) to contain calculated A
- Z (pointed by {nu1ZBase,u2Length}) to contain calculated Z
- The workspace (pointed by {nu1WorkSpace,32})

### 43.3.4.12.4. Parameters Definition

**Table 43-37.** GCD Service Parameters

| Parameter | Type | Dir. | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| Specific/Gf2n | Bit | I | – | – | GF(2n) Bit | – |
| nu1XBase | nu1 | I | Crypto RAM | u2Length | Base of X Number X | Base of X<br>Filled with the GCD D |
| u2Length | u2 | I | – | – | Length of the Areas X, Y, A, Z | Length of the Areas X, Y, A, Z |
| nu1YBase | nu1 | I | Crypto RAM | u2Length | Base of Y Number Y | Base of Y Cleared area |
| nu1ABase | nu1 | I | Crypto RAM | u2Length | Base of A | Base of A<br>Filled with the result |
| nu1ZBase | nu1 | I | Crypto RAM | u2Length + 4 (see **Note 1**) | Base of Z | Base of Z<br>Filled with the result |
| nu1WorkSpace | nu1 | I | Crypto RAM | 32 bytes | Base of the workspace | Base of the workspace corrupted |

**Note:**

1. The additional word is 4 zero bytes.

The parameters X and Y must have their most significant 32-bit word cleared to zero. The length u2Length is the length of the longer of the parameters X and Y including this zero word.

To clarify here is an example:

- X is an 8 bytes number.
- Y is a 12 bytes number.

This example is processed this way before the use of the GCD service:

- The longer number is Y so its length is taken and increased by 4 bytes for the 32-bit word cleared to zero, this gives u2Length = 16 bytes. Therefore, X, Y, A and Z areas have a length of 16 bytes.
- Y is padded with 4 bytes cleared to zero on the MSB side and the u2Length = 16 bytes are written in memory (LSB first).
- X is padded with 8 bytes cleared to zero on the MSB side and the u2Length = 16 bytes are written in memory (LSB first).
- The areas A and Z are mapped in memory with a size of u2Length = 16 bytes.
- The workspace is mapped in memory with its constant size of 32 bytes

### 43.3.4.12.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;
// Fill all the fields
PUKCL(u2Option) = 0;
PUKCL_GCD(nu1XBase) = <Base of the ram location of X>;
PUKCL_GCD(nu1YBase) = <Base of the ram location of Y>;
PUKCL_GCD(nu1ABase) = <Base of the ram location of A>;
PUKCL_GCD(nu1ZBase) = <Base of the ram location of Z>;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL_GCD(nu1WorkSpace) = <Base of the workspace>;
PUKCL_GCD(u2Length) = <Length of X, Y, A and Z>;
// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GCD, pvPUKCLParam);
if (PUKCL_Param.Status == PUKCL_OK)
                {
                // The GCD has been executed correctly
                ...
                }
else // Manage the error
```

#### 43.3.4.12.6. Constraints

The following conditions must be avoided to ensure that the service works correctly:

- nu1XBase, nu1YBase, nu1ABase or nu1ZBase are not aligned on 32-bit boundaries
- {nu1XBase, u2Length}, {nu1YBase, u2Length}, {nu1ABase, u2Length} or {nu1ZBase, u2Length} are not in Crypto RAM
- u2Length is either: < 4, > 0xffc or not a 32-bit length
- {nu1XBase, u2Length} overlaps {nu1YBase, u2Length} or {nu1XBase, u2Length} overlaps {nu1ABase, u2Length} or {nu1XBase, u2Length} overlaps {nu1ZBase, u2Length} or {nu1YBase, u2Length}overlaps

{nu1ABase, u2Length} or {nu1YBase, u2Length} overlaps {nu1ZBase, u2Length} or {nu1ABase, u2Length} overlaps {nu1ZBase, u2Length}

#### 43.3.4.12.7. Status Returned Values

**Table 43-38.** GCD Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |

### 43.3.4.13. Get Random Number

#### 43.3.4.13.1. Purpose

The purpose of this command is to provide the user with a source of entropy. The options available for this service are:

- Generation of random numbers from a Hardware Random Number Generator (TRNG).
- Generation of random numbers from a Deterministic Random Number Generator (DRNG).

**Important:**
When using this service, be sure to strictly follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG. If the directives require not to use this service, follow them and use the proposed method to get random numbers.

This service only has the option to get random numbers and does not seed, initialize or start the RNG. Other options are reserved for future use.

Neither continuous testing nor entropy testing is included in this service. If this is needed (FIPS 140, ZKA, ...), this service should not be used and the users shall develop their own command.

The DRNG is compatible with both ANSI X9.31 and FIPS 186-2 standards (see the important note below). The DRNG is designed according to:

- The algorithm described in the document ANSI Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA) X9.31 dated September 9, 1998.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The Change recommendation for ANSI X9.0 - 1995 (Part 1) and ANSI X9.31 -1998:

The algorithm B.2.1 Algorithm for computing m Values of x is the one applied in the Toolbox 3 X9.31 DRNG. The DRNG is compatible with:

- The DRNG is described in the document NIST Digital Signature Standard (DSS) FIPS Pub 186-2 January 27, 2000 Appendix 3.1
- The FIPS 186-2 Change Notice 1 dated October 5, 2001 modifies this algorithm.

**Important:** To apply the FIPS 186-2 algorithm, the parameters XSeed[0] and XSeed[1] must be set to the same value.

#### 43.3.4.13.2. How to Use the Service

#### 43.3.4.13.3. Description

This service has four possible options described in Table 43-41. Two of these options are reserved for future use. This service performs the following operations:

- Generation of a random number from the Hardware RNG
- Generation of a random number from the Deterministic RNG

**Generation of a Random Number from the Hardware RNG**

This service, activated with the option PUKCL_RNG_GET, makes it possible to get a random number R from the Hardware RNG:

R = HardwareRandomGenerate()

In the Generation of random from the RNG service, the following parameters need to be provided:

- R the generated number area (pointed by{nu1RBase,u2RLength})

#### 43.3.4.13.4. Generation of a Random Number from the Deterministic RNG

This service, activated with the option PUKCL_RNG_X931_GET, makes it possible to get a random number R from the Deterministic Random Number Generator with input parameters the Key XKey and the Seed XSeed:

(XKey, R) = DeterministicRandomGenerateFromSeed ( XKey, XSeed, Q)

In the generation of a random number from the Deterministic RNG service, the following parameters need to be provided:

- XKey the input and output Key (pointed by {nu1XKeyBase,u2XKeyLength})
- XSeed the input Seed (pointed by {nu1XseedBase,u2XKeyLength})
- Q the prime number (pointed by {nu1QBase, 20bytes})
- R the generated number area (pointed by {nu1RBase, 20bytes})

#### 43.3.4.13.5. Hardware RNG Parameters Definition

The parameters for the generation of random from the Hardware RNG are described in the following table. This service can easily be accessed through the use of the `PUKCL_Rng()` and `PUKCL()` macros.

**Table 43-39.** RNG Service Hardware Generated Parameters

| Parameter | Type | Dir. | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Option (see Table 43-41) | Option (see Table 43-41) |
| nu1RBase | nu1 | I | Crypto RAM or Device RAM | u2RLength | Base of R | Base of R filled with random values depending on the option |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-39.** RNG Service Hardware Generated Parameters (continued)

| Parameter | Type | Dir. | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2RLength | u2 | I | – | – | Length of R | Length of R |

### 43.3.4.13.6. Deterministic RNG Parameters Definition

The parameters for the generation of random from the Deterministic RNG are described in the following table. This service can easily be accessed through the use of the `PUKCL_Rng()` and `PUKCL()` macros.

**Table 43-40.** RNG Service Deterministic Generated Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Option (see Table 43-41) | Option (see Table 43-41) |
| nu1XKeyBase | nu1 | I/O | Crypto RAM | u2XKeyLength | Base of XKey | Base of XKey filled with the resulting XKey |
| nu1Workspace | nu1 | NA | Crypto RAM | 64 bytes | Base of the workspace | Base of the workspace corrupted |
| nu1Workspace2 (see **Note 1**) | nu1 | NA | Crypto RAM | 2*u1XKeyLength + 4 | Base of the workspace 2 | Base of the workspace corrupted |
| nu1XSeedBase | nu1 | I/O | Crypto RAM | max ( 2*u2XKeyLength, 44 bytes) | Base of the values XSeed[0] and XSeed[1] | Base of XSeed filled with the result on 20 bytes |
| u2XKeyLength | u2 | I | – | – | Length of XKey, Xseed[0] and Xseed[1] | Length of XKey, Xseed[0] and Xseed[1] |
| nu1QBase | nu1 | I | Crypto RAM | 20 bytes | Base of Q | Base of Q |
| nu1RBase | nu1 | I | Crypto RAM | u2RLength | Base of R | Base of R filled with the result on 20 bytes |

**Note:**

1. The nu1 Workspace2 must be a multiple of 256.

### 43.3.4.13.7. Options

The option is set by the u2Options input parameter that must take one of the values listed in the following table. Please note that the values, OPTION_RNG_SEED and OPTION_RNG_GETSEED, are reserved for future use.

**Table 43-41.** RNG Service Options

| Option | Purpose | Required Parameters |
|---|---|---|
| PUKCL_RNG_SEED | Reserved | Reserved |
| PUKCL_RNG_GET | Generation of a random number from the RNG | nu1RBase, u2RLength |
| PUKCL_RNG_X931_GET | Generation of a random number from the Deterministic RNG | nu1XKeyBase, nu1Workspace, nu1XSeedBase, u2XKeyLength, nu1QBase, nu1RBase |
| PUKCL_RNG_GETSEED | Reserved | Reserved |

### 43.3.4.13.8. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL(u2Option) =...;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
// Initializing parameters
PUKCL_Rng(nu1RBase) = <Base of the ram location to store the rng>;
PUKCL_Rng(u2RLength) = <Length of the rng to get>;

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(Rng,pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            // The RNG generation has been executed correctly
            ...
            }
else // Manage the error
```

#### 43.3.4.13.9. Constraints

**Random Number Generation**
The following conditions must be avoided to ensure that the service works correctly:

- {nu1RBase,u2RLength} not in RAM
- {nu1RBase,u2RLength} not accessible or authorized for writing

**Deterministic Random Number Generation**
The length of the parameter nu1XSeedbase is: XSeedLength = max( 2*u2XKeyLength, 44 bytes) The `max()` macro takes a maximum of two values.

The following conditions must be avoided to ensure that the service works correctly:

- nu1XKeyBase,nu1Workspace, nu1Workspace2, nu1XSeedBase, nu1QBase, nu1RBase are not aligned on 32-bit boundaries
- {nu1XKeyBase, u2XKeyLength}, {nu1Workspace, 64 bytes}, {nu1Workspace2, 2*u1XKeyLength +4}, {nu1XSeedBase, XSeedLength}, {nu1QBase, 24 bytes} or {nu1RBase, 20 bytes} are not in PUKCC RAM
- u2XKeyLength is either: < 20, > 64 or not a 32-bit length
- nu1Workspace2 not multiple of 256.
- Overlaps exist between two or more of the areas: {nu1XKeyBase, u2XKeyLength}, {nu1Workspace,64 bytes}, {nu1XSeedBase, XSeedLength}, {nu1QBase, 24 bytes} or {nu1RBase, 20 bytes}
  The area {nu1RBase, 20} can overlap with {nu1Workspace, 64 bytes} or {nu1QBas, 24 bytes}. The pointer nu1RBase can equal the pointer nu1XSeedBase.

#### 43.3.4.13.10. Status Returned Values

**Table 43-42.** RNG Service Return Codes

| Returned status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | Information | Service functioned correctly |

### 43.3.5. Modular Arithmetic Services

This section provides a complete description of the modular arithmetic services, which consists of two sets:

- Modular reductions, which can be used as stand alone operations, or used as a final step of most arithmetic operations (full and small multiplications, squaring).
- Modular operations, which include modular exponentiations (with or without using the CRT) and a probabilistic prime number generation.

These operations work on general data so the modulus has no special form. The modular services are available through:

- a Fast form (may return a congruence of the result, with a high probability to have a Normalized result)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- a Normalized form (returns the exact result, strictly lower than the modulus)
- a Euclidean form (returns the exact result, strictly lower than the modulus)

The following table describes the modes of the modular reduction with the hypothesis:

- In GF(p): The modulus is N with length NLength in bytes
- In GF($2^n$): The modulus is P[X] with length NLength in bytes

For the exact calculus of NLength see below.

**Table 43-43.** Modular Reduction Modes

| Modular Reduction Form | Input Dynamic | Result Dynamic | Comments |
|---|---|---|---|
| Fast | GF(p): 0 ≤ Input < ($N^2$) * ($2^{32}$)<br>GF(2n): Input < (($P[x]$)$^2$) * ($X^{32}$) | GF(p): 0 ≤ Res < N * 4<br>GF($2^n$): Res < P[X] * ($X^2$) | The fastest reduction available, needs a precomputed constant. |
| Normalized | InputLength < NLength + 4 bytes | GF(p): 0 ≤ Res < N GF($2^n$): Res < P[X] | The correction step does not runs in constant time. Needs a precomputed constant.<br>The Normalize function cannot be applied to the product of two numbers of length u2NLength. |
| Using Euclidean division | InputLength < 2 * NLength + 4 bytes | GF(p): 0 ≤ Res < N<br>GF($2^n$): Res < P[X] | Does not need any precomputed constant. |

To be able to use these modular reduction services (except the Euclidean division), first the implementer shall call the setup service, providing the modulus as well as one free memory space for the constant (this constant is used to speed up the modular reduction). In most commands (except the modular exponentiation), the quotient is stored in the high order bytes of the number to be reduced, using only eight bytes more than the maximum size of the number to be reduced.

The following rules must be respected to ensure the modular reduction services function correctly:

- The numbers to be reduced can have any significant length, given the fact it CANNOT BE GREATER than 2*u2ModLength + 4 bytes.
- The modulus SHALL ALWAYS HAVE a significant length of <u2ModLength> bytes. The modulus must be provided as a <u2ModLength + 4> bytes long number, padded on the most significant side with a 32-bit word cleared to zero. Not respecting this rule leads to unexpected and wrong results from the modular reduction.
- The normalization operation ALWAYS performs a modular reduction step, and will therefore have the same memory usage as this one.
- The very first operation before any modular operation SHALL BE a modular setup.

### 43.3.5.1. Modular Reduction

#### 43.3.5.1.1. Purpose

This service is used to perform the various steps necessary to perform a modular reduction and accepts as input numbers in GF(p) or polynomials in GF($2^n$) .

The available options for this service are:

- Work in the GF($2^n$) or in the standard integer arithmetic field GF(p)
- Operation is the generation of the reduction constant.
- Operation is a Modular Reduction.
- Operation is a Normalization.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.5.1.2. How to Use the Service

#### 43.3.5.1.3. Description

This service performs one of the following operations:

- Setup of the Fast or Normalize functions: generation of the reduction constant
- Fast Modular Reduction
- Big Modular Reduction (using Euclide's division)
- Normalization

The service name for this operation is `RedMod`.

#### 43.3.5.1.4. Modular Reduction Setup

This service calculates the constant Cns, computed from the modulus and used to speed up the modular reduction:

Cns = SetupConstant(N)

This service must be processed before the use of the Fast or Normalize functions. In the Setup computations, the following data must be provided:

- N the modulus (pointed by {nu1ModBase,u2ModLength +4}).
- Cns the Setup Constant Result (pointed by {nu1CnsBase,u2ModLength +12}).
- X used as a workspace (pointed by {nu1XBase,2 * u2ModLength + 8}) (include the supplementary bytes; see **Note 2** in Table 43-44
- R used as a workspace (pointed by {nu1RBase,64 or 68bytes}).
- u2ModLength is the Aligned Significant Length of the modulus and is not the byte Significant Length (see Aligned Significant Length).

#### 43.3.5.1.5. Fast Reductions and Normalization

These commands calculate an approximated or exact Modular Reduction, that is, the result may be greater than the modulus, but is always congruent to the true result.

**Important:** Before using these functions, ensure that the constant Cns has been calculated with the setup for the Modular Reduction service.

Input and Result significant values verify:

- For the Fast Modular Reduction:

$0 \leq X < N^2 \times 2^{32}$

$R = Xmod(N) + k \times N \quad with \quad 0 \leq k \leq 4$

- For the Normalize:

$XLength < (NLength + 4)bytes \;\; R$

$= Xmod(N)$

In these Fast Modular Reduction and Normalize computations, the following data have to be provided:

- X (pointed by {nu1XBase,2 * u2ModLength +8})
  - The Normalize computation accept as entry a value whose length is lower or equal to u2ModLength + 4 (that is, for example, a value yet reduced but not normalized.). The u2ModLength + 4 MSB bytes are cleared at the beginning of the computation.
  - in case of Fast RedMod computations, the value X mayverify: X < ($N^2$) *($2^{32}$).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

– include the supplementary bytes; see **Note 3** in Table 43-45)

- R (pointed by {nu1RBase,u2Modlength +4})
- N (pointed by {nu1ModBase,u2ModLength +4})
- Cns (pointed by {nu1CnsBase,u2ModLength +12})
- u2ModLength is the Aligned Significant Length of the modulus and is not the byte Significant Length (see Aligned Significant Length).

The Fast Modular Reduction is able to reduce inputs up to <2*u2ModLength + 4> bytes. The input can come from a multiplication of 2 <u2ModLength + 4> bytes numbers. The input X is considered as a <2*u2ModLength + 8> bytes number.

**Important:** Additionally the Fast Reduction and Normalize functions need supplemental bytes located on the MSB side of the number to be reduced but these bytes are restored at the end of the operation and are therefore unchanged. However, these bytes are to be taken into account when the mapping is created, and could lead to unexpected results if overlapping with other area used by the function.

The Fast Modular Reduction returns a <u2ModLength + 4> bytes number, but this number is in fact a <u2ModLength + 2> significant bytes number. When using the Fast Modular Reduction, the two MSB bytes of the <u2ModLength + 2> can have a maximum of two lsb bits set (depending on the reduced number and the modulo).

The Normalize computation accepts as entry a resulting value of Fast Modular Reduction and computes an exact result. It can not be applied to the result of the product of two numbers of size NLength: a Fast Modular Reduction must be applied before.

For the Normalize computation, the X value is limited by the preceding formula but the area in memory is bigger as described in Table 43-45.

As input, the Normalize sub-service only accept values, which length is lower or equal to u2ModLength + 4. The Most Significant u2ModLength + 4 bytes are firstly cleared by this service.

#### 43.3.5.1.6. Big Modular Reduction Using Euclide's Division

This command calculates:

$$XLength < (2 \times NLength + 4) bytes \quad R$$

$$= Xmod(N)$$

In this Big Modular Reduction computations, the following data must be provided:

- X (pointed by {nu1XBase,2 * u2ModLength + 8}) (include the supplementary bytes; see **Note 1** in Table 43-46)
- R (pointed by {nu1RBase,u2Modlength +4})
- N (pointed by {nu1ModBase,u2ModLength +4})
- u2ModLength is the Aligned Significant Length of the modulus and is not the byte Significant Length (see Aligned Significant Length)
- Workspace (pointed by {nu1CnsBase,64 or 68}).

#### 43.3.5.1.7. Modular Reductions Service Parameters Definition

**Table 43-44.** RedMod Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-44.** RedMod Service Parameters (continued)

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| Specific/CarryIn | Bits | I | – | – | Must be set to zero. | – |
| Specific/Gf2n | Bit | I | – | – | GF($2^n$) Bit | – |
| Specific/CarryOut Zero Violation | Bits | I | – | – | – | Carry Out, Zero Bit and Violation Bit filled according to the result |
| nu1ModBase ( see **Note 1**) | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of N | Base of N untouched |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 12 | Base of Cns | Base of Cns filled with the Setup Constant |
| u2ModLength | u2 | I | – | – | Length of N | Length of N |
| nu1RBase | nu1 | I | Crypto RAM | GF(p): 64 bytes<br>GF(2n): 68 bytes | Base of R<br>as a workspace | Base of R workspace corrupted |
| nu1XBase (see **Note 2**) | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Base of X as a workspace | Base of X workspace corrupted |

**Notes:**

1. The Modulus is to be given as a u2ModLength Aligned Significant Length Bytes however, it has to be provided as a u2ModLength + 4 bytes long number, having the four high-order bytes set to zero.
2. Before the X (pointed by {nu1XBase,2 * u2ModLength + 8}) LSB bytes, four supplementary bytes will be saved/restored. Other four supplementary bytes will also be saved/restored after the X MSB bytes. All these supplementary bytes may be entirely in the Crypto RAM (therefore, do not place the X area too near the end of the Crypto RAM) and shall not overlap with other area used by the service.

### 43.3.5.1.8. Fast Modular Reductions Service Parameters Definition

**Table 43-45.** Fast RedMode and Normalize Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |
| Specific/CarryIn | Bits | I | – | – | Must be set to zero. | – |
| Specific/Gf2n | Bit | I | – | – | GF(2n) Bit | – |
| Specific/CarryOut Zero Violation | Bits | I | – | – | – | Carry Out, Zero Bit and Violation Bit filled according to the result |
| nu1ModBase (see **Note 1**) | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of N | Base of N untouched |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 12 | Base of Cns | Base of Cns untouched |
| u2ModLength | u2 | I | – | – | Length of N | Length of N |
| nu1RBase (see **Note 2**) | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of R | Base of R filled with the result |
| nu1XBase (see **Note 3**) | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Base of X the number to reduce | Base of X corrupted |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. The Modulus is to be given as a u2ModLength Aligned Significant Length Bytes however, it has to be provided as a u2ModLength + 4 bytes long number, having the four high-order bytes set to zero.
2. To make profitable the space memory, it is possible to set nu1RBase exactly equal to nu1XBase.
3. After the X (pointed by {nu1XBase,2 * u2ModLength + 8}) MSB bytes, supplementary bytes will be saved/restored (8 bytes in case of Fast RedMod, otherwise; 12 bytes). These supplementary bytes may be entirely in the Crypto RAM (therefore, do not place the X area too near the end of the Crypto RAM) and shall not overlap with other area used by the service.

#### 43.3.5.1.9. Big Modular Reduction Parameters Definition

**Table 43-46.** Big RedMod Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |
| Specific/CarryIn | Bits | I | – | – | Must be set to zero | – |
| Specific/Gf2n | Bit | I | – | – | GF(2n) Bit | – |
| Specific/CarryOut Zero Violation | Bits | I | – | – | – | Carry Out, Zero Bit and Violation Bit filled according to the result |
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of N | Base of N untouched |
| nu1CnsBase | nu1 | I | Crypto RAM | GF(p): 64 bytes<br>GF(2n): 68 bytes | Base of Cns as a workspace | Base of Cns corrupted |
| u2ModLength | u2 | I | – | – | Length of N | Length of N |
| nu1RBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of R | Base of R filled with the result |
| nu1XBase (see **Note 1**) | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Base of X the number to reduce | Base of X filled with the result |

**Note:**

1. Before the X (pointed by {nu1XBase,2 * u2ModLength + 8}) LSB bytes, four supplementary bytes will be saved/restored. Other four supplementary bytes will also be saved/restored after the X MSB bytes. All of these supplementary bytes may be entirely in the Crypto RAM (therefore, do not place the X area too near the end of the Crypto RAM) and shall not overlap with other area used by the service.

#### 43.3.5.1.10. Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Operation Option described in Table 43-47
- if the Operation Option is PUKCL_REDMOD_REDUCTION, the Modular Reduction Sub-Option described in Table 43-48

The u2Options number is calculated by an Inclusive OR of the options. Some Examples in C language are:

- Operation: Setup for the ModularReductions.
  ```
  PUKCL(u2Options) = PUKCL_ REDMOD_SETUP;
  ```
- Operation: Fast ModularReduction.
  ```
  PUKCL(u2Options) = PUKCL_REDMOD_REDUCTION | PUKCL_REDMOD_USING_FASTRED;
  ```

For this command three exclusive options can be specified. The following table lists the operations that can be performed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-47.** RedMod Service Options

| Option | Purpose | Required Parameters |
|---|---|---|
| PUKCL_REDMOD_SETUP | Perform the Cns value computation | nu1ModBase, u2ModLength, nu1CnsBase, nu1XBase |
| PUKCL_REDMOD_REDUCTION | Perform R ≡ X Mod N, see sub-option for details | nu1ModBase, u2ModLength, nu1CndBase, nu1XBase, nu1RBase |
| PUKCL_REDMOD_NORMALIZE | Perform R = X Mod N | nu1ModBase, u2ModLength, nu1CndBase, nu1XBase, nu1RBase |

When selecting the PUKCL_REDMOD_REDUCTION option, one of the two sub-options listed in the following table must be selected.

**Table 43-48.** RedMode Service Options with PUKCL_RED_MOD_REDUCTION

| Option | Purpose | Required Parameters |
|---|---|---|
| PUKCL_REDMOD _USING_DIVISION | Perform R = X Mod N | nu1ModBase, u2ModLength, nu1CndBase, nu1XBase |
| PUKCL_REDMOD _USING_FASTRED | Perform R ≡ X Mod N<br>The entropy is minimized (~2 bits) | nu1ModBase, u2ModLength, nu1CndBase, nu1XBase, nu1RBase |

### 43.3.5.1.11. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

PUKCL(Specific).CarryIn = 0;
PUKCL(Specific).GF2n = ...;

PUKCL(u2Option) =...;

// Depending on the option specified, not all fields should be filled
PUKCL_RedMod(nu1ModBase) = <Base of the ram location of N>;
PUKCL_RedMod(u2ModLength) = <Length of N>;
PUKCL_RedMod(nu1CnsBase) = <Base of the ram location of Cns>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(RedMod,pvPUKCLParam);
if (PUKCL_Param.Status == PUKCL_OK)
            {
            // operation has correctly been performed
            ...
            }
else // Manage the error
```

### 43.3.5.1.12. Constraints

Depending on the options chosen the lengths of the R area and Cns area differ:

- For the Setup:
  - RLength = 64bytes
  - CnsLength = u2ModLength +12
- For the Fast Reduction and Normalize:
  - RLength = u2ModLength +4
  - CnsLength = u2ModLength +8
- For the BigRedMod:
  - RLength = u2ModLength +4
  - CnsLength =64

The following combinations of input values should be avoided in the case of a modular reduction ‘alone’, meaning that it has not been requested as an option of any other command:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- nu1ModBase, nu1CnsBase, nu1RBase, nu1XBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2CnsLength}, {nu1XBase, 2*u2XLength + 8 + s} or {nu1RBase, u2RLength} are not in Crypto RAM
- u2ModLength is either: < 4, > 0xffc or not a 32-bit length
- Overlaps exist between two or more of the areas: {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2CnsLength}, {nu1XBase, 2*u2XLength + 8 + s} or {nu1RBase, u2RLength}

**Note:** Overlaps between {nu1RBase, RLength} and {nu1XBase, 2*u2XLength + 8} are forbidden; but if the operation is the Fast, Normalized or Big Modular Reduction, the equality between nu1RBase and nu1XBase is authorized.

#### 43.3.5.1.13. Status Returned Values

**Table 43-49.** RedMod Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |
| PUKCL_DIVISION_BY_ZERO | Severe | When computing an Euclidean division, the Modulus was found to be zero. This occurs ONLY when either reducing using an Euclidean division or computing the reduction constant usable for a Fast or Normalize Reduction. |
| PUKCL_UNEXPLOITABLE_OPTIONS | Severe | A bad combination of options has been detected. |
| PUKCL_MALFORMED_MODULUS | Severe | The Msw of the modulus is not zero. |

### 43.3.5.2. Modular Exponentiation (Without CRT)

#### 43.3.5.2.1. Purpose

This service is used to perform the Modular Exponentiation computation. This service processes integers in GF(p) only.

The options available for this service are:

- Fast implementation
- Regular implementation
- Exponent is located in Crypto RAM or not in Crypto RAM
- Exponent window size

#### 43.3.5.2.2. How to Use the Service

#### 43.3.5.2.3. Description

**Important:** Before using these functions, ensure that the constant Cns has been calculated with the Setup of the Modular Reductions service.

This service processes the following operation:

The service name for this operation is `ExpMod`.

$R = X^{Exp} mod(N)$

In this computation, the following parameters need to be provided:

- X: input number (pointed by {nu1XBase,u2ModLength +16})
- N: modulus (pointed by {nu1ModBase,u2ModLength +4}).
- Exp: exponent (pointed by {pfu1ExpBase,u2ExpLength +4})
- Cns: Fast Modular Constant (pointed by {nu1CnsBase,u2ModLength +8})


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Precomp: precomputation workspace (pointed by{nu1PrecompBase,PrecompLen})
- Blinding: exponent blinding value (provided inu1Blinding)

The length PrecompLen depends on the lengths and options chosen; its calculus is detailed in Options below.

**Note:** The minimum value for u2ModLength is 12 bytes. Therefore, the significant length of N must be at least three 32-bit words.

#### 43.3.5.2.4. Parameters Definition

**Table 43-50.** ExpMod Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of N | Base of N untouched |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns untouched |
| u2ModLength | u2 | I | – | – | Length of N | Length of N |
| nu1XBase (see **Note 1**) | nu1 | I | Crypto RAM | u2ModLength + 16 | Base of X | Base of X<br>Filled with the result |
| nu1PrecompBase | nu1 | I | Crypto RAM | See below | Base of Precomp as a workspace | Base of Precomp workspace corrupted |
| pfu1ExpBase (see **Note 2**) | pfu1 | I | Any place (see **Note 3**) | u2ExpLength + 4 | Base of the Exponent | Base of the Exponent untouched |
| u2ExpLength (see **Note 4**) | u2 | I | – | – | Significant length of Exponent | Significant length of Exponent |
| u1Blinding (see **Note 5**) | u1 | I | – | – | Exponent unblinding value | Exponent unblinding value untouched |

**Notes:**

1. This zone contains the number to be exponentiated (u2ModLength bytes) and is used during the computations as a workspace (four 32-bit words longer than the number to be exponentiated). At the end of the computation, it contains the correct result of the operation.
2. The exponent must be given with a supplemental word on the LSB side (low addresses). This word shall be set to zero.
3. If the PUKCL_EXPMOD_EXPINPUKCCRAM option is not set, the location of the exponent MUST NOT be the Crypto RAM, even partially.
4. The u2ExpLength parameter does not take into account the supplemental word needed on the LSB side of the exponent.
5. It is possible to mask the exponent in memory using an 8-bits XOR mask value. Be aware that not only the exponent, but also the supplemental word has to be masked. If masking is not desired, then this parameter should be set to 0.

#### 43.3.5.2.5. Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Calculus Mode Option described in Table 43-51
- the mandatory Window Size Option described in Table 43-52
- the indication of the presence of the exponent in Crypto RAM

**Note:** Please check precisely if one part of the exponent is in Crypto RAM. If this is the case the PUKCL_EXPMOD_EXPINPUKCCRAM must be used.

The u2Options number is calculated by an “Inclusive OR” of the options. Some examples in C language are:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Operation:Fast Modular Exponentiation with the window size equal to 1 and with no part of the Exponent in the Crypto RAM
  `PUKCL(u2Options) = PUKCL_EXPMOD_FASTRSA | PUKCL_EXPMOD_WINDOWSIZE_1;`
- Operation: Regular Modular Exponentiation with the window size equal to 2 and with one part of the Exponent in the Crypto RAM
  `PUKCL(u2Options) = PUKCL_EXPMOD_REGULARRSA | PUKCL_EXPMOD_WINDOWSIZE_2 | PUKCL_EXPMOD_EXPINPUKCCRAM;`

There is no difference on the final result when using any of the options for this service. The choice has to be made according to the available resources (RAM, Time) and also taking into account the expected security level.

For this service, two exclusive Calculus Modes are possible. The following table describes the Calculus Mode Options.

**Table 43-51.** ExpMod Service Calculus Mode Option

| Option | Explanation |
|---|---|
| PUKCL_EXPMOD_FASTRSA | Performs a Fast computation |
| PUKCL_EXPMOD_REGULARRSA | Performs a Regular computation, slower than the Fast version, but using Regular calculus methods |

For this service, four window sizes are possible. The window size in bits is those of the windowing method used for the exponent.

The choice of the window size is a balance between the size of the parameters and the computation time:

- Increasing the window size increases the precomputation workspace.
- Increasing the window size reduces the computation time (may not be relevant for very small exponents).

The following table details the size of the precomputation workspace, depending on the chosen window size option.

**Table 43-52.** ExpMode Service Window Size Options and Precomputation Space Size

| Option specified | Size of the PrecompBase Workspace (bytes) | Content of the Workspace |
|---|---|---|
| PUKCL_EXPMOD_WINDOWSIZE_1 | 3*(u2ModLength + 4) + 8 | x |
| PUKCL_EXPMOD_WINDOWSIZE_2 | 4*(u2ModLength + 4) + 8 | $x\ x^3$ |
| PUKCL_EXPMOD_WINDOWSIZE_3 | 6*(u2ModLength + 4) + 8 | $x\ x^3\ x^5\ x^7$ |
| PUKCL_EXPMOD_WINDOWSIZE_4 | 10*(u2ModLength + 4) + 8 | $x\ x^3\ x^5\ x^7\ x^9\ x^{11}\ x^{13}\ x^{15}$ |

The exponent can be located in RAM or in the data space. If one part of the exponent is in Crypto RAM this must be mandatory signaled by using the option PUKCL_EXPMOD_EXPINPUKCCRAM.

The following table describes this option.

**Table 43-53.** ExpMod Service Exponent in Crypto RAM Option

| Option | Purpose |
|---|---|
| PUKCL_EXPMOD_EXPINPUKCCRAM | The exponent can be read from any data space of memory, including Flash, RAM or even Crypto RAM. When at least one word the exponent is in Crypto RAM, this option has to be set. |

### 43.3.5.2.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL(u2Option) =...;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
// Depending on the option specified, not all fields should be filled
PUKCL_ExpMod(nu1ModBase) = <Base of the ram location of N>;
PUKCL_ExpMod(u2ModLength) = <Length of N>;
PUKCL_ExpMod(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL_ExpMod(nu1XBase) = <Base of the ram location of X>;
PUKCL_ExpMod(nu1PrecompBase) = <Base of the ram location of Precomp>;
PUKCL_ExpMod(pfu1ExpBase) = <Base of the location of Exp>;
PUKCL_ExpMod(u2ExpLength) = <Length of Exp>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ExpMod, pvPUKCLParam);
if (PUKCL_Param.Status == PUKCL_OK)
            {
            // operation has been performed correctly
            ...
            }
else // Manage the error
```

#### 43.3.5.2.7. Constraints

The following combinations of input values should be avoided in the case of a modular reduction 'alone', meaning that it has not been requested as an option of any other command:

- nu1ModBase,nu1CnsBase, nu1XBase,nu1PrecompBase,nu1ExpBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1XBase, u2ModLength +16},{nu1PrecompBase, <PrecompLength>} are not in Crypto RAM
- {nu1ExpBase,u2ExpLength + 4} has no part in Crypto RAM and PUKCL_EXPMOD_EXPINPUKCCRAM is specified
- u2ModLength or u2ExpLength are either: < 4, > 0xffc or not a 32-bit length
- None or both PUKCL_EXPMOD_REGULARRSA and PUKCL_EXPMOD_FASTRSA are specified.
- {nu1PrecompBase,<PrecompLength>} overlaps with either: {nu1ModBase, u2ModLength +4}, {nu1CnsBase, u2ModLength + 8} {nu1XBase, u2ModLength + 16} or {nu1ExpBase, u2ExpLength + 4}
- {nu1XBase,u2ModLength + 16} overlaps with either: {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8} or {nu1ExpBase, u2ExpLength + 4}
- {nu1ModBase, u2ModLength + 4} overlaps {nu1CnsBase, u2ModLength +8}

#### 43.3.5.2.8. Maximum Sizes for the Modular Exponentiation

The following table provides the maximum sizes for the Modular Exponentiation, depending on the window size and the presence of the exponent in Crypto RAM.

- The figures below are calculated supposing that u2ExpLength =u2ModLength.
- In case of the PUKCL_EXPMOD_EXPINPUKCCRAM option is specified, for the computation of the maximum acceptable size, it is assumed the Exponent is entirely in the Crypto RAM and its length is equal to the Modulus one.
- Otherwise, the Exponent is entirely out of the Crypto RAM and so the computation do not depend on its length.

**Table 43-54.** Maximum Exponentiation Sizes

| Option Specified | Maximum Modulus Size (bytes) | Maximum Modulus Size (bits) |
|---|---|---|
| Exponent in Crypto RAM, 1 bit window | 576 | 4608 |
| Exponent in Crypto RAM, 2 bits window | 504 | 4032 |
| Exponent in Crypto RAM, 3 bits window | 400 | 3200 |
| Exponent in Crypto RAM, 4 bits window | 284 | 2272 |
| Exponent not in Crypto RAM, 1 bit window | 672 | 5376 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-54.** Maximum Exponentiation Sizes (continued)

| Option Specified | Maximum Modulus Size (bytes) | Maximum Modulus Size (bits) |
|---|---|---|
| Exponent not in Crypto RAM, 2 bits window | 576 | 4608 |
| Exponent not in Crypto RAM, 3 bits window | 448 | 3584 |
| Exponent not in Crypto RAM, 4 bits window | 308 | 2464 |

#### 43.3.5.2.9. Status Returned Values

**Table 43-55.** ExpMod Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | Service functioned correctly |

### 43.3.5.3. Probable Prime Generation (Using Rabin-Miller)

#### 43.3.5.3.1. Purpose

This service is used to perform probable prime generation or test. This service processes integers in GF(p) only.

The options available for this service are:

- Choice of the number of iterations of the Rabin-Miller test
- Generation or Test of a probable prime number
- Fast Implementation
- Regular Implementation
- Exponent Window Size

#### 43.3.5.3.2. Additional Information

The Rabin-Miller test is a probable-primality testing algorithm. As a consequence, the primality of the generated number is not guaranteed at 100%, however, numerous publications have been issued explaining how to estimate the probability of getting a composite number, giving the size of the number and the number of iterations (the T parameter).

Useful information can be found in the *"Handbook of Applied Cryptography (Discrete Mathematics and Its Applications" by Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone*, in the following sections:

- 4.2.3. "Rabin-Miller Test"
- 4.4. "Prime Number Generation"

#### 43.3.5.3.3. How to Use the Service

#### 43.3.5.3.4. Description

This service processes a test for probable primality or a generation of a probable prime number.

**Note:** When using this service be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.

This service processes one of the following operations: CheckProbablePrimality(N)

or

N = GenerateProbablePrimeFromSeed (NSeed)

In this computation, the following parameters need to be provided:

- N the input number (pointed by {nu1NBase,u2NLength +4})
  - If the requested operation is a test, it is untouched after the operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - If the requested operation is a generation and a probable prime number was found before reaching the Maximum Increment, it contains the resulting probable prime after the operation.
  - If the requested operation was a generation and Maximum Increment was reached before a probable prime number was found, it contains no relevant information.

- Cns as a workspace (pointed by {nu1CnsBase,u2NLength +12})
- Rnd as a workspace (pointed by {nu1RndBase,u2NLength +16})
- Precomp the precomputation workspace (pointed by{nu1PrecompBase,PrecompLen})
- Exp as a workspace (pointed by {pfu1ExpBase,u2ExpLength +4})
- u1MillerRabinIterations the number of Miller Rabin Iterations requested
- u2MaxIncrement, maximum increment of the number in case of probable prime generation

The length PrecompLen depends on the lengths and options chosen; its calculus is detailed in Options below.

The service name for this operation is `PrimeGen`.

#### 43.3.5.3.5. Parameters Definition

**Table 43-56.** PrimeGen Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1NBase (see **Note 1**) | nu1 | I | Crypto RAM | u2NLength + 4 | Base of N<br>Number to test or Seed for the generation | Base of N unchanged if test or generation result ( see **Note 1**) |
| nu1CnsBase | nu1 | I | Crypto RAM | u2NLength + 12 | Base of Cns as a workspace | Base of Cns workspace corrupted |
| u2NLength | u2 | I | – | – | Length of N | Length of N |
| nu1RndBase | nu1 | I | Crypto RAM | Max (u2NLength + 16,64) | Internal Workspace | Internal Workspace corrupted |
| nu1PrecompBase | nu1 | I | Crypto RAM | See Options below | Base of Precomp workspace | Base of Precomp workspace corrupted |
| nu1RBase (see **Note 2**) | nu1 | – | Crypto RAM | – | – | – |
| nu1ExpBase (see **Note 3**) | nu1 | I | Crypto RAM | u2NLength + 4 | Base of Exponent (R) | Base of Exponent (R) |
| u1MillerRabin-Iterations | u1 | I | – | – | Miller Rabin's T parameter | Miller Rabin's T parameter |
| u2MaxIncrement | u2 | I | – | – | Maximum Increment (see **Note 4**) | Maximum Increment |

**Notes:**

1. This zone contains the number to be either tested or used as a seed for generation. It has to be provided with one zero word on the MSB side. This area has supplementary constraints (see the following Important note).

1. This parameter does not have to be provided and is used as an internal value for computing the reduction's constant.
2. The area {nu1ExpBase, u2NLength + 4} must be entirely in the Crypto RAM.
3. The generation starts from the number in {nu1NBase,u2NLength + 4} and increments it until a number is found as probable prime. However, the generation may stop for two reasons: The number has been incremented in a way it is bigger than <u2NLength> bytes, or the original number has been incremented by more than <u2MaxIncrement>.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

In case of probable prime generation, ensure that the addition of NSeed and Maximum Increment is not a number with more bytes than u2NLength, as this would produce an overflow.

**Important:**
One additional word is used on the LSB side of the NBase parameter; this word is restored at the end of the calculus. As a consequence, the parameter nu1NBase must never be at the beginning of the Crypto RAM, but at least at one word from the beginning.

One additional word is used on the MSB side of the NBase parameter; this word is not corrupted. As a consequence the Area {nu1NBase, u2NLength} must not be at the end of the Crypto RAM but at least at one word from the end.

Prime numbers of a size lower than 96 bits (three 32-bit words) cannot be generated or tested by this service.

#### 43.3.5.3.6. Options

Some of the Prime Generation options configure the Modular Exponentiation steps and so are very similar to the Modular Exponentiation options.

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Operation Option described in Table 43-57
- the mandatory Calculus Mode Option described in Table 43-58
- the mandatory Window Size Option described in Table 43-59

The u2Options number is calculated by an “Inclusive OR” of the options. Some Examples in C language are:

- Operation: Probable Prime Testing with Fast Modular Exponentiation and the window size equal to 1
```
PUKCL(u2Options) = PUKCL_PRIMEGEN_TEST | PUKCL_EXPMOD_FASTRSA |
PUKCL_EXPMOD_WINDOWSIZE_1;
```
- Operation: Probable Prime Generate with Regular Modular Exponentiation and the window size equal to 2
```
PUKCL(u2Options) = PUKCL_EXPMOD_REGULARRSA | PUKCL_EXPMOD_WINDOWSIZE_2;
```

The following table describes the PrimeGen service features available from the various options.

**Table 43-57.** PrimeGen Service Options

| Option | Method Used |
|---|---|
| PUKCL_PRIMEGEN_TEST | This option is used to specify that only tests will be made on the provided number.<br>When this option is not specified, a prime generation algorithm is selected, starting from the given seed and incrementing it. |
| PUKCL_EXPMOD_WINDOWSIZE_1,2,3 or 4 | Depending on this option, different bit-window sizes will be used. For long exponents, the bigger the window, the faster the computation. However, this has also an impact on the size of the precomputations table. |

For this service, two exclusive Calculus Modes are possible. The following table describes the Calculus Mode Options.

**Table 43-58.** PrimeGen Service Calculus Mode Options

| Option | Explanation |
|---|---|
| PUKCL_EXPMOD_FASTRSA | Perform a Fast computation. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-58.** PrimeGen Service Calculus Mode Options (continued)

| Option | Explanation |
|---|---|
| PUKCL_EXPMOD_REGULARRSA | Performs a Regular computation, slower than the Fast version, but using regular calculus methods. |

The length of the Precomp area depends on the window size W and u2NLength. The Precomp area length is:

PrecompLen = max( 2*(u2NLength + 4) + 2W-1 * (u2NLength + 4), u2NLength + 8 + 64) + 8

**Note:** Please calculate precisely the length PrecompLen with the formula and the `max()` macro, which takes a maximum of two values.

The following table shows the size of the precomputation workspace (PrecompLen), depending on the chosen window size option.

**Table 43-59.** PrimeGen Service Precomputation Space Size

| Option Specified | Size of the PrecompBase Workspace (bytes) | Content of the Workspace |
|---|---|---|
| PUKCL_EXPMOD_WINDOWSIZE_1 | max( 3*(u2NLength + 4), u2NLength + 72) + 8 | x |
| PUKCL_EXPMOD_WINDOWSIZE_2 | max( 4*(u2NLength + 4), u2NLength + 72) + 8 | $x\ x^3$ |
| PUKCL_EXPMOD_WINDOWSIZE_3 | max( 6*(u2NLength + 4), u2NLength + 72) + 8 | $x\ x^3\ x^5\ x^7$ |
| PUKCL_EXPMOD_WINDOWSIZE_4 | max( 10*(u2NLength + 4) u2NLength + 72) + 8 | $x\ x^3\ x^5\ x^7\ x^9\ x^{11}\ x^{13}\ x^{15}$ |

The following table provides the maximum sizes for the Prime Generation depending on the window size.

**Table 43-60.** PrimeGen Service Maximum Sizes

| Characteristics of the Operation | Maximum Prime Sizes (bits) |
|---|---|
| 1 bit window | 4608 |
| 2 bits window | 4032 |
| 3 bits window | 3200 |
| 4 bits window | 2272 |

### 43.3.5.3.7. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip PUKCL(u2Option) =...;
// Depending on the option specified, not all fields should be filled
PUKCL_PrimeGen(nu1NBase) = <Base of the ram location of N>;
PUKCL_PrimeGen(u2NLength)  =  <Length  of  N>;
PUKCL_PrimeGen(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL_PrimeGen(nu1PrecompBase) = <Base of the ram location of Precomp>;
PUKCL_PrimeGen(pfu1ExpBase) = <Base of the location of Exp>;
PUKCL_PrimeGen(u2ExpLength) = <Length of Exp>;
PUKCL_PrimeGen(u1MillerRabinIterations) = <Number of iterations>;
PUKCL_PrimeGen(u2MaxIncrement) = <Maximum Increment>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(PrimeGen, pvPUKCLParam);
if (PUKCL_Param.Status == PUKCL_NUMBER_IS_PRIME)
            {
            // The number is probably prime
            ...
            }
else if (PUKCL_Param.Status == PUKCL_NUMBER_IS_NOT_PRIME)
            {
            // The number is not prime
            ...
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
                }
else // Manage the error
```

#### 43.3.5.3.8. Constraints

The following combinations of input values should be avoided in the case of a modular reduction 'alone', meaning that it has not been requested as an option of any other service:

- nu1NBase,nu1CnsBase, nu1RndBase,nu1PrecompBase,nu1ExpBase are not aligned on 32-bit boundaries
- {nu1NBase, u2NLength + 4}, {nu1CnsBase, u2NLength + 12}, {nu1RndBase, u2NLength +12}, {nu1PrecompBase, <PrecompLength>} are not in Crypto RAM
- u2NLength is either: < 12, > 0xffc or not a 32-bit length
- Both PUKCL_EXPMOD_REGULARRSA and PUKCL_EXPMOD_FASTRSA are specified.
- {nu1PrecompBase,<PrecompLength>} overlaps with either: {nu1NBase, u2NLength + 4}, {nu1CnsBase, u2NLength + 12} {nu1RndBase, u2NLength + 12} or {nu1ExpBase, u2ExpLength + 4}
- {nu1RndBase,3*u2NLength + 24} overlaps with either: {nu1NBase, u2NLength + 4},{nu1CnsBase, u2NLength + 12} {nu1XBase, u2NLength + 12} or {nu1ExpBase, u2ExpLength + 4}
- {nu1NBase, u2NLength + 4} overlaps {nu1CnsBase, u2NLength +12}

#### 43.3.5.3.9. Status Returned Values

**Table 43-61.** PrimeGen Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_NUMBER_IS_PRIME | Information | The generated or tested number has been detected as probably prime. |
| PUKCL_NUMBER_IS_NOT_PRIME | Information | The generated or tested number has been detected as composite. |

### 43.3.5.4. Modular Exponentiation (With CRT)

#### 43.3.5.4.1. Purpose

The purpose of this service is to perform the Modular Exponentiation with the Chinese Remainders Theorem (CRT). This service processes integers in GF(p) only.

The options available for this service are:

- Fast implementation
- Regular implementation
- Exponent is located in Crypto RAM or not
- Exponent window size

#### 43.3.5.4.2. How to Use the Service

#### 43.3.5.4.3. Description

This service processes a Modular Exponentiation with the Chinese Remainder Theorem:

$R = X^{D} mod(N)$ *with* $N = P * Q$

**Important:** For this service, be sure to follow the directives given for the RSA implementation on the chip you use.

This service requires that the modulus N is the product of two co-primes P and Q and that the decryption exponents D is co-prime with the product ((P-1)*(Q-1)).

The Input data are P, Q, EP, EQ, Rvalue, and X. P and Q are the co-primes so that N = P*Q.

X is the number to exponentiate.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

EP, EQ and Rval are calculated as follows:

$EP = Dmod(P - 1)$ $EQ = Dmod(Q - 1)$ $Rval = P^{-1}mod(Q)$

In some cases, the decryption exponent D may not be available and the encryption exponent E may be available instead. The possibilities to calculate the parameters are:

- Calculate D from E with the formula:
  $D = E^{-1}mod((P - 1) \times (Q - 1))$
- Calculate the parameters from E:
  $EP = E^{-1}mod(P - 1)$ $EQ = E^{-1}mod(Q - 1)$ $Rval = P^{-1}mod(Q)$

In this computation, the following parameters need to be provided:

- X the input number (pointed by {nu1XBase,2*u2ModLength +16})
- P and Q the primes (pointed by {nu1ModBase,2*u2ModLength +8}).
- EP and EQ the reduced exponents (pointed by {pfu1ExpBase,2*u2ExpLength +8})
- Rval and Precomp (pointed by{nu1PrecompBase,RAndPrecompLen})
- Blinding the exponent blinding value (provided inu1Blinding)

The length RAndPrecompLen depends on the lengths and options chosen; its calculus is detailed in Options below.

The service for this operation is CRT.

**Note:** The minimum value for u2ModLength is 12 bytes. Therefore, the significant length of P or Q must be at least three 32-bit words.

#### 43.3.5.4.4. Parameters Definition

The following table shows the parameter block for the CRT service.

Many parameters have complex placement in memory; therefore, detailed figures are provided in CRT Service Placement below.

**Table 43-62.** CRT Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Options | u2 | I | – | – | Options (see below) | Options (see below) |
| nu1ModBase | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Base of P, Q | Base of P, Q untouched |
| u2ModLength | u2 | I | – | – | Length of P or Q greater than or equal to 12 | Length of P or Q |
| nu1XBase (see **Note 1**) | nu1 | I | Crypto RAM | 2*u2ModLength + 16 | Base of X | Base of X<br>Filled with the result |
| nu1PrecompBase | nu1 | I | Crypto RAM | See Options below | Base of Rvalue and Pre computations workspace | Corrupted |
| pfu1ExpBase (see **Note 2**) | pfu1 | I | Any place | 2*u2ExpLength + 8 | Base of EP, EQ | Base of EP, EQ untouched |
| u2ExpLength | u2 | I | – | – | Significant length of EP or EQ | Significant length of EP or EQ |
| u1Blinding (see **Note 3**) | u4 | I | – | – | Exponent unblinding value | Exponent unblinding value |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. This zone contains the number to be exponentiated (u2ModLength bytes) and is used during the computations as a workspace (four 32-bit words longer than the number to be exponentiated). At the end of the computation, it contains the correct result of the operation.
2. If the PUKCL_EXPMOD_EXPINPUKCCRAM option is not set, the location of the exponent MUST NOT be placed in the Crypto RAM, even partially.
3. It is possible to mask the exponent in memory using a 32-bit XOR mask value. Be aware that not only the exponent, but also the supplemental spill word has to be masked. If masking is not desired, the parameter should be set to 0.

#### 43.3.5.4.5. Options

Most of the CRT options configure the Modular Exponentiation steps of the CRT and so are very similar to the Fast Modular Exponentiation options.

The options are set by the u2Options input parameter, which is composed of:

- the mandatory Calculus Mode Option described in Table 43-63
- the mandatory Window Size Option described in Table 43-64
- the indication of the presence of the exponent in Crypto RAM

**Important:** Please check precisely if one part of the exponent area (containing EP and EQ) is in Crypto RAM. If this is the case, the PUKCL_EXPMOD_EXPINPUKCCRAM option must be used.

The u2Options number is calculated by an “Inclusive OR” of the options. Some Examples in C language are:

- Operation: CRT using the Fast Modular Exponentiation with the window size equal to 1 and with no part of the Exponent area in the Crypto RAM
  ```
  PUKCL(u2Options) = PUKCL_EXPMOD_FASTRSA | PUKCL_EXPMOD_WINDOWSIZE_1;
  ```
- Operation:CRT using the Regular Modular Exponentiation with the window size equal to 2 and with one part the Exponent area in the Crypto RAM
  ```
  PUKCL(u2Options) = PUKCL_EXPMOD_REGULARRSA | PUKCL_EXPMOD_WINDOWSIZE_2 |
  PUKCL_EXPMOD_EXPINPUKCCRAM;
  ```

For this service, two exclusive Calculus Modes for the Modular Exponentiation steps of the CRT are possible. The following table describes the Calculus Mode Options.

**Table 43-63.** CRT Service Calculus Mode Options

| Option | Explanation |
|---|---|
| PUKCL_EXPMOD_FASTRSA | Perform a Fast computation. |
| PUKCL_EXPMOD_REGULARRSA | Performs a Regular computation, slower than the Fast version, but using regular calculus methods. |

For this service, four window sizes for the Modular Exponentiation Steps are possible. The window size in bits is those of the windowing method used for the exponent.

The choice of the window size is a balance between the size of the parameters and the computation time:

- Increasing the window size increases the precomputation workspace.
- Increasing the window size reduces the computation time (may not be relevant for very small exponents). The length of the Rval and Precomp area depends on the window size W and u2ModLength.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The Rval and Precomp area length is:

RandPrecompLen = 4 * (u2ModLength + 4) + max(64 , $2^{(W-1)}$ * (u2ModLength + 4)) + 8

**Important:** Please calculate precisely the length RandPrecompLen with the formula and the `max()` macro, which takes the maximum of two values.

The following table shows the size of the Rval and Precomp area, depending on the chosen window size option.

**Table 43-64.** CRT Service Window Size Options and Rval and Precomp Area Size

| Option Specified | Size of the Rval and Precomp Area (bytes) | Precomputation Values |
|---|---|---|
| PUKCL_EXPMOD_WINDOWSIZE_1 | 4*(u2ModLength + 4) + max(64 , (u2ModLength + 4)) + 8 | x |
| PUKCL_EXPMOD_WINDOWSIZE_2 | 4*(u2ModLength + 4) + max(64 , 2*(u2ModLength + 4)) + 8 | x $x^3$ |
| PUKCL_EXPMOD_WINDOWSIZE_3 | 4*(u2ModLength + 4) + max(64 , 4*(u2ModLength + 4)) + 8 | x $x^3$ $x^5$ $x^7$ |
| PUKCL_EXPMOD_WINDOWSIZE_4 | 10*(u2ModLength + 4) + max(64 , 8*(u2ModLength + 4)) + 8 | x $x^3$ $x^5$ $x^7$ $x^9$ $x^{11}$ $x^{13}$ $x^{15}$ |

The exponent area can be located in RAM or in the data space. If one part of the exponent area is in Crypto RAM this must be mandatory signaled by using the PUKCL_EXPMOD_EXPINPUKCCRAM option.

The following table describes this option.

**Table 43-65.** CRT Service Crypto RAM Option Exponent Area

| Option | Purpose |
|---|---|
| PUKCL_EXPMOD_EXPINPUKCCRAM | The exponent area can be read from any data space of memory, including Crypto RAM. When at least one word the exponent is in Crypto RAM, this option has to be set. |

### 43.3.5.4.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL(u2Option) =...;

// Depending on the option specified, not all fields should be filled PUKCL_CRT(nu1ModBase) =
<Base of the ram location of P and Q>; PUKCL_CRT(u2ModLength) = <Length of P or Q>;
PUKCL_CRT(nu1XBase) = <Base of the ram location of X>;
PUKCL_CRT(nu1PrecompBase) = <Base of the ram location of RVal and Precomp>;
PUKCL_CRT(pfu1ExpBase) = <Base of the ram location of EP and EQ>;
PUKCL_CRT(u2ExpLength) = <Length of EP or EQ>;
PUKCL_CRT(u1Blinding) = <Blinding value>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(CRT, pvPUKCLParam);
if (PUKCL_Param.Status == PUKCL_OK)
            {
            // operation has been performed correctly
            ...
            }
else // Manage the error
```

### 43.3.5.4.7. Constraints

The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1XBase, nu1PrecompBase, pfu1ExpBase are not aligned on 32-bit boundaries
- {nu1XBase, 2*u2ModLength + 16}, {nu1ModBase, 2*u2ModLength + 8}, {nu1PrecompBase,<PrecompLength>} are not in Crypto RAM


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- {nu1ExpBase,2*u2ExpLength + 8} is not in Crypto RAM and PUKCL_EXPMOD_EXPINPUKCCRAM is specified
- u2ModLength or u2ExpLength are either: < 4, > 0xffc or not a 32-bit length
- None or both PUKCL_EXPMOD_REGULARRSA and PUKCL_EXPMOD_FASTRSA are specified.
- {nu1XBase,2*u2ModLength + 16} overlaps with either: {nu1ModBase, 2*u2ModLength +8}, {nu1PrecompBase, <PrecompLength>} or {pfu1ExpBase, 2*u2ExpLength + 8}
- {nu1ModBase,2*u2ModLength + 8} overlaps with either: {nu1PrecompBase, <PrecompLength>} or {pfu1ExpBase, 2*u2ExpLength + 8}
- {nu1PrecompBase, <PrecompLength>} overlaps {pfu1ExpBase, 2*u2ExpLength +8}

#### 43.3.5.4.8. CRT Service Parameter Placement

The parameters' placements are described in detail in the following figures.

**Figure 43-2.** Modulus P and Q in {nu1ModBase, 2*u2ModLength + 8}

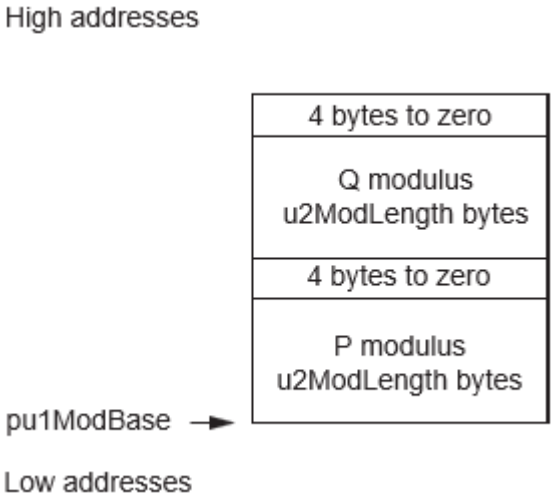
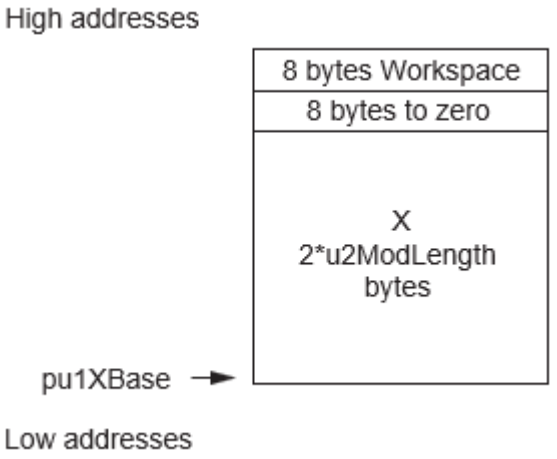



**Figure 43-3.** Value X in {nu1XBase, 2*u2ModLength + 16}





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 43-4.** Exponents EP and EQ in {fnu1ExpBase, 2*u2ExpLength + 8}

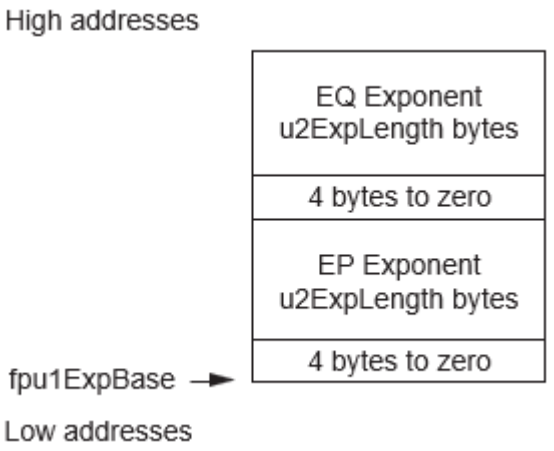
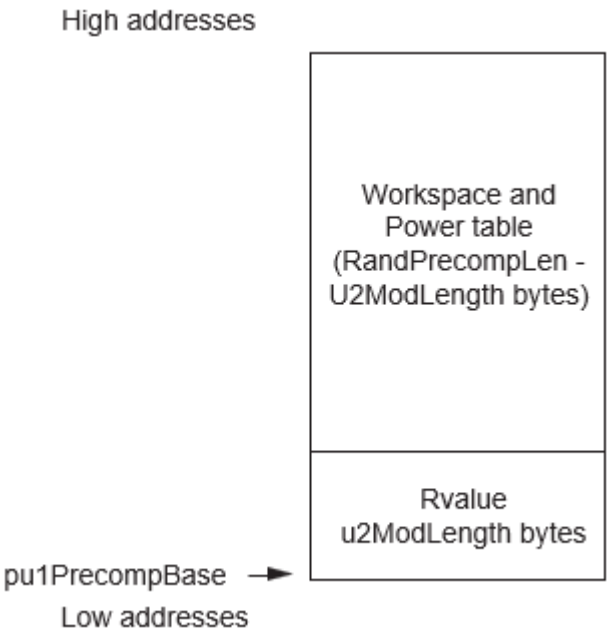



**Figure 43-5.** Value Rval and Precomp in {nu1PrecompBase, RandPrecompLen}




### 43.3.5.4.9. CRT Service Modular Exponentiation Maximum Size

The following table details the maximum size in bits of P or Q, of N and of EP or EQ.

- The maximum size in bits of P or Q equals:
  <Max Size Bits P> = <Max Size Bits Q> = 8 * <Max u2ModLength bytes>
- The maximum size in bits of N=P*Q equals:
  <Max Size Bits N> = 2 * <Max Size Bits P>
- The maximum size in bits of EP or EQ equals:
  <Max Size Bits EP> = <Max Size Bits EQ> = 8 * <Max u2ExpLength bytes>
- In case of the PUKCL_EXPMOD_EXPINPUKCCRAM option is specified, for the computation of the maximum acceptable size, it is assumed the Exponent is entirely in the Crypto RAM and its length equal the Modulus one.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Otherwise, the Exponent is entirely out of the Crypto RAM and so the computation do not depend on its length.

**Table 43-66.** CRT Service Maximum Sizes

| Characteristics of the Operation | P or Q Max Bit Sizes | N Max Bit Sizes | EP or EQ Max Bit Sizes |
|---|---|---|---|
| Exponent in Crypto RAM, 1 bit window | 2912 | 5824 | 2912 |
| Exponent in Crypto RAM, 2 bits window | 2688 | 5376 | 2688 |
| Exponent in Crypto RAM, 3 bits window | 2464 | 4928 | 2464 |
| Exponent in Crypto RAM, 4 bits window | 2304 | 4608 | 2304 |
| Exponent not in Crypto RAM, 1 bit window | 3584 | 7168 | <application dependent> |
| Exponent not in Crypto RAM, 2 bits window | 3232 | 6464 | <application dependent> |
| Exponent not in Crypto RAM, 3 bits window | 2912 | 5824 | <application dependent> |
| Exponent not in Crypto RAM, 4 bits window | 2688 | 5376 | <application dependent> |

#### 43.3.5.4.10. Status Returned Values

**Table 43-67.** CRT Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | Information | Service functioned correctly |

### 43.3.6. Elliptic Curves Over GF(p) Services

This section provides a complete description of the currently available elliptic curve over Prime Fields services. These services process integers in GF(p) only.

The offered services cover the basic operations over elliptic curves such as:

- Adding two points over a curve
- Doubling a point over a curve
- Multiplying a point by an integral constant
- Converting a point's projective coordinates (resulting from a doubling or an addition) to the affine coordinates, and oppositely converting a point's affine coordinates to the projective coordinates.
- Testing the point presence on the curve.

Additionally, some higher level services covering the needs for signature generation and verification are offered:

- Generating an ECDSA signature (compliant with FIPS186-2)
- Verifying an ECDSA signature (compliant with FIPS186-2) The supported curves use the following curve equation:

$Y^2 = X^3 + aX + b$

#### 43.3.6.1. Coordinate Systems

##### 43.3.6.1.1. General Considerations

In this implementation, several choices have been made related to the coordinate systems managed by the elliptic curve primitives.

There are two systems currently managed by the library:

- Affine Coordinates System where each curve point has two coordinates (X, Y)
- Projective Coordinates System where each point is represented with three coordinates (X,Y, Z)

Converting from the affine coordinates system to a projective coordinates system is performed by extending its representation with Z = 1:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

$(X, Y) \Rightarrow (X, Y, Z= 1)$

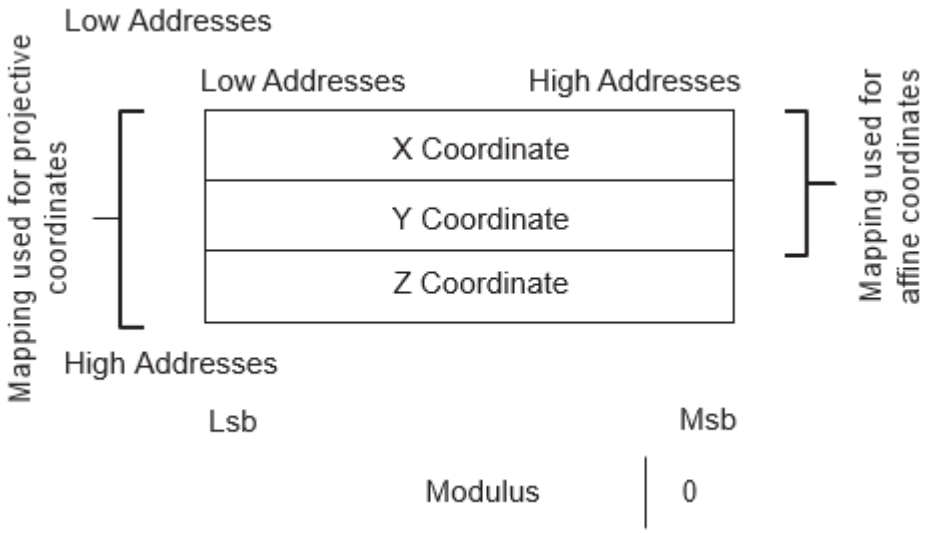
Converting from a projective coordinate to an affine one is a service offered by the PUKCL. The formula to perform this conversion is:

$(X, Y, Z) \Rightarrow (X / Z^2, Y / Z^3)$

#### 43.3.6.1.2. Points Representations

Depending on the representation (Projective or Affine), points are represented tn memory, as shown in the following figure.

**Figure 43-6.** Points Representation in Memory




In this figure, the modulus is represented as a reference, and to show that coordinates are always to be provided on the length of the modulus plus one 32-bit word.

The different types of representations are as follows:

$$\text{Affine representation} \qquad Pt = \begin{bmatrix} X_{Affine} < P \times 2^{15} \\ Y_{Affine} < P \times 2^{15} \end{bmatrix}$$

$$\text{Projective representation} \qquad Pt = \begin{bmatrix} X_{Projective} < P \times 2^{15} \\ Y_{Projective} < P \times 2^{15} \\ Z_{Projective} < P \times 2^{15} \end{bmatrix}$$

**Notes:**

1. The minimum value for u2ModLength is 12 bytes. Therefore, the significant length of the modulus must be at least three 32-bit words.
2. In some cases the point can be the infinite point. In this case, it is represented with its Z coordinates equal or congruent to zero.

#### 43.3.6.1.3. Modulus and Modular Constant Parameters

In most of the services the following parameters must be provided:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- P the Modulus (often pointed by {nu1ModBase,u2ModLength + 4}): This parameter contains the Modulus Integer prime P defining the Galois Field used in points coordinates computations. The Modulus must be u2ModLength bytes long, while having a supplemental zeroed 32-bit word on the MSB side.
  **Note:** Most of the Elliptic Curve computations are reduced modulo P. In many functions the reductions are made with the Fast Reduction.
- Cns the Modular Constant (often pointed by {nu1CnsBase,u2ModLength + 12}): This parameter contains the Modular Constant associated to the Modulus

**Important:** The Modular Constant must be calculated before using the GF(p) Elliptic Curves functions by a call to the Setup for Modular Reductions with the GF(p) option (see Modular Reduction Setup in the Modular Reduction section).

#### 43.3.6.2. Point Addition

##### 43.3.6.2.1. Purpose

This service is used to perform a point addition, based on a given elliptic curve over GF(p). Please note that:

- This service is not intended to add the same point twice. In this particular case, use the doubling service
  (see Fast Point Doubling).

##### 43.3.6.2.2. How to Use the Service

##### 43.3.6.2.3. Description

The operation performed is:

$Pt_C = Pt_A + Pt_B$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase,3*u2ModLength + 12}). This point can be the Infinite Point.
- B the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointBBase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 5*u2ModLength +32}

The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the very same place than the input point A. This Point can be the Infinite Point.

The service name for this operation is `ZpEccAddFast`. This service uses Fast mode and Fast Modular Reduction for computations.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the Setup of the Modular Reduction functions.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.6.2.4. Parameters Definition

**Table 43-68.** ZpEccAddFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of Modulus P | Base of Modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulo | Length of modulo |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (projective coordinates) | Resulting point C (projective coordinates) |
| nu1PointBBase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point B (projective coordinates) | Input point B |
| nu1Workspace | nu1 | I | Crypto RAM | 5*u2ModLength + 32 | – | Corrupted workspace |

#### 43.3.6.2.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

PUKCL (u2Option) = 0;

PUKCL _ZpEccAdd(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEccAdd(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEccAdd(u2ModLength) = <Byte length of P>;
PUKCL _ZpEccAdd(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEccAdd(nu1PointBBase) = <Base of the ram location of the B point>;
PUKCL _ZpEccAdd(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccAddFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.6.2.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1PointBBase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PointBBase, 3*u2ModLength + 12}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8},{nu1PointABase, 3*u2ModLength + 12}, {nu1PointBBase, 3*u2ModLength + 12} and {nu1Workspace, 5*u2ModLength + 32}

#### 43.3.6.2.7. Status Returned Values

**Table 43-69.** ZpEccAddFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 43.3.6.3. Point Addition and Subtraction

#### 43.3.6.3.1. Purpose

This service is used to perform a point addition and point subtraction, based on a given elliptic curve over GF(p). Please note that:

- This service is not intended to add the same point twice. In this particular case, use the doubling service (see Fast Point Doubling).

#### 43.3.6.3.2. How to Use the Service

#### 43.3.6.3.3. Description

The operation performed is:

$Pt_C = Pt_A \pm Pt_B$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase,3*u2ModLength + 12}). This point can be the Infinite Point.
- B the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointBBase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 5*u2ModLength +32}
- The operator filled with the operation to perform (Addition or Subtraction)

The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the very same place than the input point A. This Point can be the Infinite Point.

The service name for this operation is `ZpEccAddSubFast`. This service uses Fast mode and Fast Modular Reduction for computations.

**Note:** Before using this service, ensure that the constant Cns has been calculated with the setup of the modular reduction functions.

#### 43.3.6.3.4. Parameters Definition

**Table 43-70.** ZpEccAddSubFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of Modulus P | Base of Modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulo | Length of modulo |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (projective coordinates) | Resulting point C (projective coordinates) |
| nu1PointBBase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point B (projective coordinates) | Input point B |
| u2Operator | u2 | I | - | - | Addition or Subtraction | Addition or Subtraction |
| nu1Workspace | nu1 | I | Crypto RAM | 5*u2ModLength + 32 | – | Corrupted workspace |

#### 43.3.6.3.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL (u2Option) = 0;

PUKCL _ZpEccAddSub(nu1ModBase) = <Base of the ram location of P>;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL _ZpEccAddSub(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEccAddSub(u2ModLength) = <Byte length of P>;
PUKCL _ZpEccAddSub(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEccAddSub(nu1PointBBase) = <Base of the ram location of the B point>;
PUKCL _ZpEccAddSub(nu1Workspace) = <Base of the ram location of the workspace>;
PUKCL _ZpEccAddSub(u2Operator) = <Operation to perform (PUKCL_ZPECCADD or PUKCL_ZPECCSUB)>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccAddSubFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
        {
        ...
        }
else // Manage the error
```

#### 43.3.6.3.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1PointBBase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PointBBase, 3*u2ModLength + 12}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8},{nu1PointABase, 3*u2ModLength + 12}, {nu1PointBBase, 3*u2ModLength + 12} and {nu1Workspace, 5*u2ModLength + 32}

#### 43.3.6.3.7. Status Returned Values

**Table 43-71.** ZpEccAddFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.6.4. Fast Point Doubling

#### 43.3.6.4.1. Purpose

This service is used to perform a Point Doubling, based on a given elliptic curve over GF(p).

#### 43.3.6.4.2. How to Use the Service

#### 43.3.6.4.3. Description

These two services process the Point Doubling:

$Pt_C = 2 \times Pt_A$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 4*u2ModLength +28}
- The a parameter relative to the elliptic curve (pointed by {nu1ABase,u2ModLength +4})
- The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the same location than the input point A. This point can be the Infinite Point.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The service name for this operation is `ZpEccDblFast`. This service uses Fast mode and Fast Modular Reduction for computations.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reduction service.

#### 43.3.6.4.4. Parameters Definition

**Table 43-72.** ZpEccDblFastService

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1ABase | u2 | I | Crypto RAM | u2ModLength + 4 | Parameter a of the elliptic curve | Parameter a of the elliptic curve |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (projective coordinates) | Resulting point C (projective coordinates) |
| nu1Workspace | nu1 | I | Crypto RAM | 4*u2ModLength + 28 | – | Corrupted workspace |

#### 43.3.6.4.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL (u2Option) = 0;

PUKCL _ZpEccDbl(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEccDbl(u2ModLength) = <Byte length of P>;
PUKCL _ZpEccDbl(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEccDbl(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEccDbl(nu1ABase) = <Base of the a parameter of the elliptic curve>;
PUKCL _ZpEccDbl(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccDblFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.6.4.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1ABase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1ABase, u2ModLength + 4} and {nu1Workspace, 4*u2ModLength + 28}


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.6.4.7. Status Returned Values

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.6.5. Fast Multiplying by a Scalar Number of a Point

#### 43.3.6.5.1. Purpose

This service is used to multiply a point by an integral constant K on a given elliptic curve over GF(p).

#### 43.3.6.5.2. How to Use the Service

#### 43.3.6.5.3. Description

These two services process the Multiplying by a scalar number:

$Pt_C = K \times Pt_A$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 8*u2ModLength +44}
- The a parameter relative to the elliptic curve (pointed by {nu1ABase,u2ModLength +4})
- K the scalar number (pointed by {nu1ScalarNumber,u2ScalarLength +4})

The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the very same place than the input point A. This point can be the Infinite Point.

The service name for this operation is `ZpEccMulFast`. This service uses Fast mode and Fast Modular Reduction for computations.

**Note:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reduction service.

#### 43.3.6.5.4. Parameters Definition

**Table 43-73.** ZpEccMulFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1KBase | nu1 | I | Crypto RAM | u2KLength | Scalar number used to multiply the point A | Unchanged |
| u2KLength | u2 | I | – | – | Length of scalar K | Length of scalar K |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (projective coordinates) | Resulting point C (projective coordinates) |
| nu1ABas | nu1 | I | Crypto RAM | u2ModLength + 4 | Parameter a of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 8*u2ModLength + 44 | – | Corrupted workspace |

#### 43.3.6.5.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL (u2Option) = 0;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL _ZpEccMul(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEccMul(u2ModLength) = <Byte length of P>;
PUKCL _ZpEccMul(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEccMul(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEccMul(nu1ABase) = <Base of the ram location of the parameter A of the elliptic
curve>;
PUKCL _ZpEccMul(nu1KBase) = <Base of the ram location of the scalar number>;
PUKCL _ZpEccMul(nu1Workspace) = <Base of the ram location of the workspace>;
PUKCL_ZpEccMul(u2KLength) = <Byte length of the Scalar Number K>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccMulFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.6.5.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase,nu1CnsBase, nu1PointABase, nu1ABase, nu1ScalarNumber, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1ABase, u2ModLength + 4}, {nu1ScalarNumber, u2ScalarLength} or {nu1Workspace, 8*u2ModLength + 44} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1ABase, u2ModLength + 4}, {nu1ScalarNumber, u2ScalarLength} and {nu1Workspace, 8*u2ModLength + 44}

#### 43.3.6.5.7. Status Returned Values

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.6.6. Quick Dual Multiplying by Two Scalar Numbers and Two Points

#### 43.3.6.6.1. Purpose

This service is used to multiply two points by two integral constants K1 and K2, and then provide the addition of these multiplications results.

**Important:** This service has a quick implementation without additional security.

#### 43.3.6.6.2. How to Use the Service

#### 43.3.6.6.3. Description

This service processes the dual Multiplying by two scalar numbers:

$PtC = K_1 \times Pt_A + K_2 \times Pt_B$

In this computation, the following parameters need to be provided:

- A the first input point is filled in projective coordinates (X,Y,Z) (pointed by {pu1PointABase, (3*(u2ModLength + 4)) * (2(WA-2))}). This point can be the Infinite Point.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- B the 2nd input point is filled in projective coordinates (X,Y,Z) (pointed by {pu1PointBBase, (3*(u2ModLength + 4)) * (2(WB-2))}). This point can be the Infinite Point.
- P the modulus filled and Cns the Fast Modular Constant filled (pointed by {pu1ModCnsBase,2*u2ModLength + 16})
- The a parameter filled and the workspace not initialized (pointed by {pu1AWorkBase, 9*u2ModLength +48}
- KAB the scalar numbers (pointed by {pu1KABBase, 2*u2KLength +8})
- The options are set by the u2Options input parameter, which is composed of:
  - wA: Size of window for Point A between 2 and15
  - wB: Size of window for Point B between 2 and15
  - PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM flag: to set only if the scalars are entirely in Classic RAM with no part in PUKCC RAM

The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at (pu1AWorkBase + u2ModLength + 4). This point can be the Infinite Point.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reduction service.

#### 43.3.6.6.4. Parameters Definition

WA is the Point A window size and WB is the Point B window size (see Options below for details).

**Important:** Please calculate precisely the length of areas with the formulas. Ensure that the pu1 type is a pointer on 4 bytes and contains the full address (see Aligned Significant Length ).

**Table 43-74.** ZpEccQuickDualMulFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| pu1ModCnsBase | pu1 | I | Crypto RAM | 2 * u2ModLength + 16 | Base of modulus P, Base of Cns | Base of modulus P, Base of Cns |
| u2Option | u2 | I | – | – | Option related to the called service (see below) | – |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| pu1KABBase | pu1 | I | Any RAM | 2 * u2KLength + 8 | Scalar numbers used to multiply the points A and B | Unchanged |
| u2KLength | u2 | I | – | – | Length of scalars KA and KB | Length of scalars KA and KB |
| pu1PointABase | pu1 | I/O | Crypto RAM | (3*(u2ModLength + 4)) * ($2^{(WA-2)}$) (1) | Input point A (projective coordinates) | Unchanged |
| pu1PointBBase | pu1 | I | Crypto RAM | (3*(u2ModLength + 4)) * ($2^{(WB-2)}$) (2) | Input point B (projective coordinates) | Unchanged |
| pu1AWorkBase | pu1 | I | Crypto RAM | 9*u2ModLength + 48 | Parameter a of the elliptic curve | Resulting point C (projective coordinates) in pu1AWorkBase Base + u2ModLength + 4 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. The precalculus table size for the point A is calculated from chosen window size “WA”.
2. The precalculus table size for the point B is calculated from chosen window size “WB”.

#### 43.3.6.6.5. Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory windows sizes WA and WB
- the indication of the presence of the scalars in system RAM

**Note:** Please check precisely if one part of the scalars is in Crypto RAM. If this is the case, the PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM option must not be used.

The u2Options number is calculated by an “Inclusive OR” of the options. Some Examples in C language are:

- ```
  // Scalars are in system RAM
  // The Point A window size is 3
  // The Point B window size is 4
  PUKCL(u2Options) = PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM |
  PUKCL_ZPECCMUL_WINSIZE_A_VAL_TO_OPT(3) |
  PUKCL_ZPECCMUL_WINSIZE_B_VAL_TO_OPT(4);
  ```
- ```
  // Scalars are in the PUKCC Cryptographic RAM
  // The Point A window size is 2
  // The Point B window size is 5
  PUKCL(u2Options) = PUKCL_ZPECCMUL_WINSIZE_A_VAL_TO_OPT(2) |
  PUKCL_ZPECCMUL_WINSIZE_B_VAL_TO_OPT(5);
  ```

For this service, many window sizes are possible. The window sizes in bits are those of the windowing method used for the scalar multiplying.

The choice of the window sizes is a balance between the size of the parameters and the computation time:

- Increasing the window size increases the precomputation table size.
- Increasing the window size to the optimum reduces the computation time.

The following table details the size of the point and the precomputation table, depending on the chosen window size option.

**Table 43-75.** ZpEccQuickDualMulFast Service Window Size Options and Precomputation Table Size

| Option Specified | Size of the Point and the Precomputation Table |
|---|---|
| PUKCL_ZPECCMUL_WINSIZE_A_VAL_TO_OPT(WA) WA in [2, 15] | (3*(u2ModLength + 4)) * $(2^{(WA-2)})$ |
| PUKCL_ZPECCMUL_WINSIZE_B_VAL_TO_OPT(WB) WB in [2, 15] | (3*(u2ModLength + 4)) * $(2^{(WB-2)})$ |

The scalars can be located in PUKCC RAM or in system RAM. If both scalars are entirely in system RAM with no part in PUKCC RAM this can be signaled by using the option PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM. In all other cases this option must not be used.

The following table describes this option.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-76.** ZpEccQuickDualMulFast Service System RAM Scalar Options

| Option | Purpose |
|---|---|
| PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM | The scalars can be located in Crypto RAM or in system RAM.<br>If both scalars are entirely in system RAM with no part in Crypto RAM this can be signaled by using this option . In all other cases this option must not be used. |

#### 43.3.6.6.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

PUKCL(u2Option) = <Configure scalar numbers location and windows sizes>;
PUKCL_ZpEccQuickDualMulFast(pu1ModCnsBase) = <Base of the ram location of P and Cns>;
PUKCL_ZpEccQuickDualMulFast(u2ModLength) = <Byte length of P>;
PUKCL_ZpEccQuickDualMulFast(u2KLength) = <Byte length of scalars>;
PUKCL_ZpEccQuickDualMulFast(pu1PointABase) = <Base of the ram location of the A point>;
PUKCL_ZpEccQuickDualMulFast(pu1PointBBase) = <Base of the ram location of the B point>;
PUKCL_ZpEccQuickDualMulFast(pu1AWorkBase) = <Base of the ram location of the parameter A of
the elliptic curve and workspace>;
PUKCL_ZpEccQuickDualMulFast(pu1KABBase) = <Base of the ram location of the scalar numbers KA
and KB>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccQuickDualMulFast, pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.6.6.7. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- pu1ModCnsBase,pu1PointABase, pu1PointBBase, pu1AWorkBase, pu1KABBase are not aligned on 32-bit boundaries
- {pu1ModCnsBase, 2*u2ModLength + 16}, {pu1PointABase, (3*(u2ModLength + 4)) *($2^{(WA-2)}$)}, {pu1PointBBase, (3*(u2ModLength + 4)) * ($2^{(WB-2)}$)} or { pu1AWorkBase, 9*u2ModLength + 48} are not in PUKCC RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- Alloverlapping between {pu1ModCnsBase, 2*u2ModLength + 16}, {pu1PointABase, (3*(u2ModLength + 4)) * ($2^{(WA-2)}$)}, {pu1PointBBase, (3*(u2ModLength + 4)) * ($2^{(WB-2)}$)} or {pu1AWorkBase, 9*u2ModLength + 48}.

#### 43.3.6.6.8. Parameters Placement

The parameters' placement is described in the following figures.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 43-7.** Modulus P and Cns{pu1ModCnsBase, 2*u2ModLength + 16}

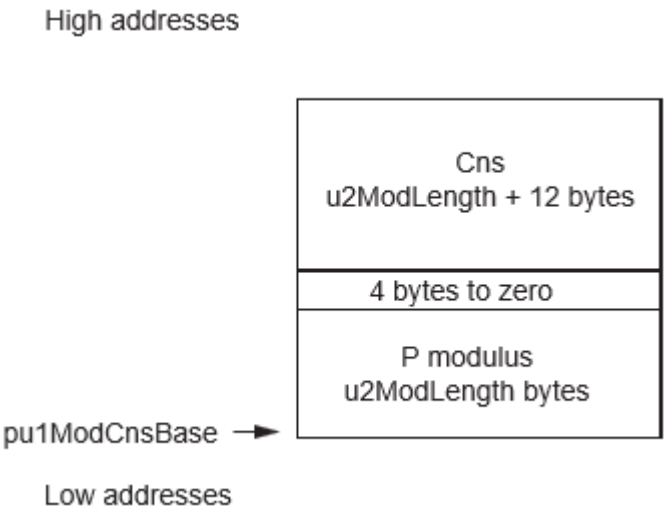
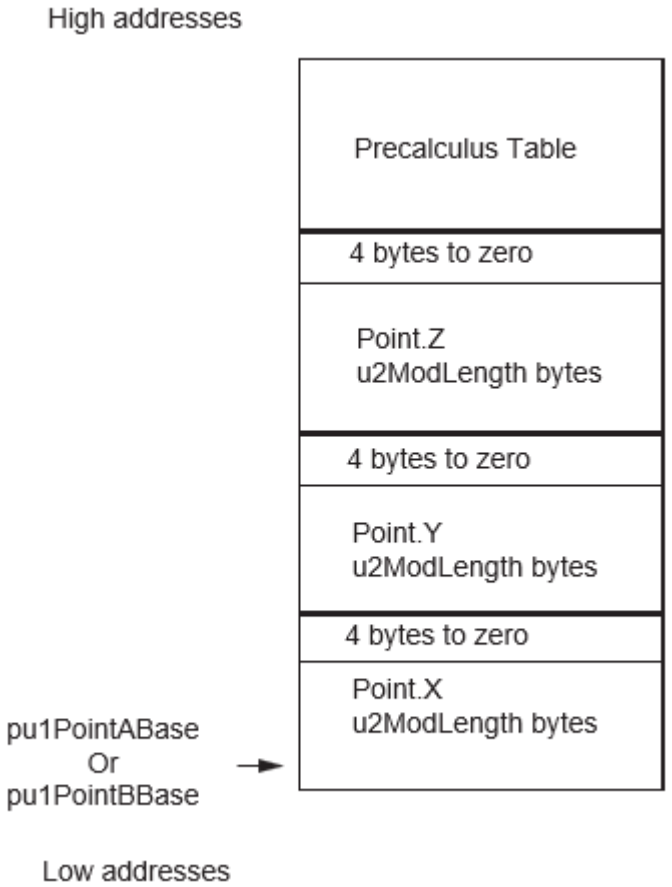



**Figure 43-8.** Points A and B {pu1PointABase, [(3*(u2ModLength + 4)) * ($2^{(WA-2)}$)] Or [(3*(u2ModLength + 4)) * ($2^{(WB-2)}$)]}





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 43-9.** Scalars KA and KB {pu1KABBase, 2 * u2KLength + 8}

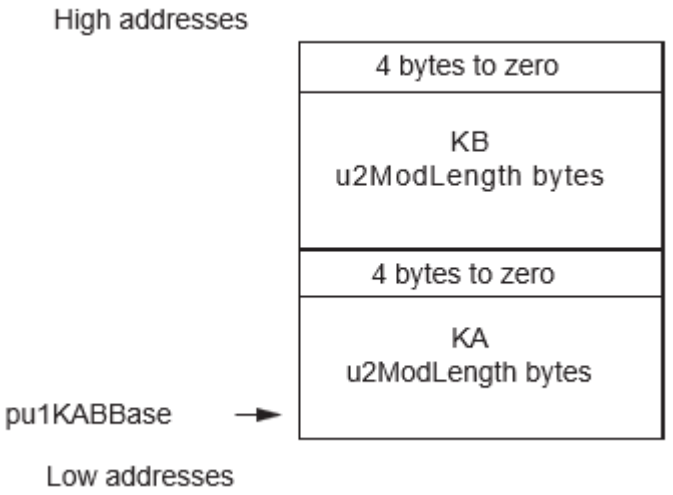
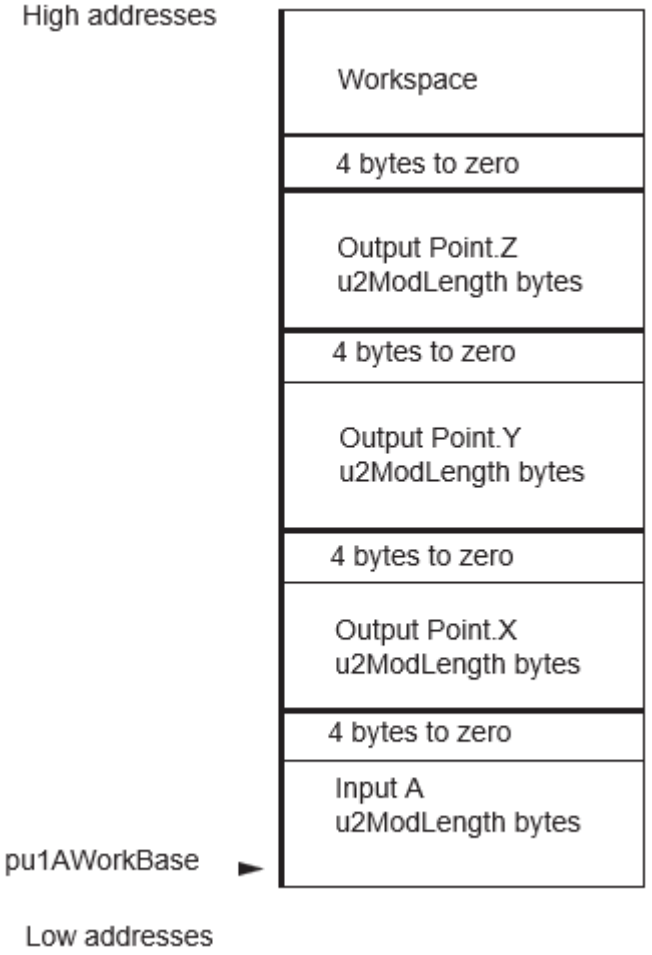



**Figure 43-10.** The a parameter and Workspace {pu1AWorkBase, 9*u2ModLength + 48}





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.6.6.9. Status Returned Values

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.6.7. Projective to Affine Coordinates Conversion

#### 43.3.6.7.1. Purpose

This service is used to perform a point coordinates conversion from projective representation to affine.

#### 43.3.6.7.2. How to Use the Service

#### 43.3.6.7.3. Description

The operation performed is:

$$Pt_{X\ Affine\ \ coordinate} = \left[\frac{Pt_{X\text{Projective coordinate}}}{\left(Pt_{Z\ \text{Projective coordinate}}\right)2}\right]$$

$$Pt_{Y\ Affine\ \ coordinate} = \left[\frac{Pt_{Y\ \text{Projective coordinate}}}{\left(Pt_{Z\ \text{Projective coordinate}}\right)3}\right]$$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) or affine coordinates for X and Y, and setting Z to 1(pointed by {nu1PointABase,3*u2ModLength + 12}). The Point A can be the point at infinity. In this case, the u2Status returned is PUKCL_POINT_AT_INFINITY.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 4*u2ModLength +48}

The result is the point A with its (X,Y) coordinates converted to affine, and the Z coordinate set to 1. The service for this operation is ZpEcConvProjToAffine.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the Setup of the fast Modular Reductions service.

#### 43.3.6.7.4. Parameters Definition

**Table 43-77.** ZpEccConvAffineToProjective Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointABase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point A | Resulting point A in affine coordinates |
| nu1Workspace | nu1 | I | Crypto RAM | 4*u2ModLength + 48 | – | Workspace |

#### 43.3.6.7.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL (u2Option) = 0;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL _ZpEcConvProjToAffine(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEcConvProjToAffine(u2ModLength) = <Byte length of P>;
PUKCL _ZpEcConvProjToAffine(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEcConvProjToAffine(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEcConvProjToAffine(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEcConvProjToAffine,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.6.7.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8},{nu1PointABase, 3*u2ModLength+ 12}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8},

{nu1PointABase, 3*u2ModLength + 12} and {nu1Workspace, 4*u2ModLength + 48}

#### 43.3.6.7.7. Status Returned Values

**Table 43-78.** ZpEccConvAffineToProjective Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |
| PUKCL_POINT_AT_INFINITY | Warning | The input point has its Z equal to zero, so it's a representation of the infinite point. |

### 43.3.6.8. Affine to Projective Coordinates Conversion

#### 43.3.6.8.1. Purpose

This service is used to perform a point coordinates conversion from an affine point representation to projective.

#### 43.3.6.8.2. How to Use the Service

#### 43.3.6.8.3. Description

The operation performed is:

*affine*(*Xa*, *Ya*) → *projective*(*Xp*, *Yp*, *Zp*)

In this computation, the following parameters need to be provided:

- A the input point is filled in affine coordinates for X and Y, and setting Z to 1 (pointed by {nu1PointABase,3*u2ModLength + 4}).
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 2*u2ModLength +16}

The result is the point A with its (X,Y,Z) projective coordinates.

The service for this operation is ZpEcConvAffineToProjective


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reductions service.

#### 43.3.6.8.4. Parameters Definition

**Table 43-79.** ZpEccConvAffineToProjective Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointABase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point A | Resulting point A in affine coordinates |
| nu1Workspace | nu1 | I | Crypto RAM | 2*u2ModLength + 16 | – | Workspace |

#### 43.3.6.8.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL (u2Option) = 0;

PUKCL _ZpEcConvAffineToProjective(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEcConvAffineToProjective(u2ModLength) = <Byte length of P>;
PUKCL _ZpEcConvAffineToProjective(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEcConvAffineToProjective(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _ZpEcConvAffineToProjective(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEcConvAffineToProjective,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.6.8.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, and {nu1Workspace, 2*u2ModLength + 16}

#### 43.3.6.8.7. Status Returned Values

**Table 43-80.** ZpEccConvAffineToProjective Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 43.3.6.9. Randomize a Coordinate

#### 43.3.6.9.1. Purpose

This service is used to convert the projective representation of a point to another projective representation.

#### 43.3.6.9.2. How to Use the Service

#### 43.3.6.9.3. Description

The operation performed is:

*Projective*($X_1$, $Y_1$, $Z_1$) → *Projective*($X_2$, $Y_2$, $Z_2$)

In this computation, the following parameters need to be provided:

- The input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointBase,3*u2ModLength + 12}). This Point must not be the point at infinity.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 3*u2ModLength +28}
- The random number (pointed by {nu1RandomBase, u2ModLength +4}).

The result is the point nu1PointBase with its (X,Y,Z) coordinates randomized.

The service for this operation is ZpEcRandomiseCoordinate.

**Important:** Before using this service:

- Ensure that the constant Cns has been calculated with the setup of the Modular Reduction service.
- Be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG

.

#### 43.3.6.9.4. Parameters Definition

**Table 43-81.** ZpEccRandomiseCoordinate Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointBase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point | Resulting point |
| nu1RandomBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Random | Corrupted |
| nu1Workspace | nu1 | I | Crypto RAM | 3*u2ModLength + 28 | – | Workspace |

#### 43.3.6.9.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL (u2Option) = 0;

// Depending on the option specified, not all fields should be filled
PUKCL _ZpEccRandomiseCoordinate(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEccRandomiseCoordinate(u2ModLength) = <Byte length of P>;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL _ZpEccRandomiseCoordinate(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL_ZpEccRandomiseCoordinate(nu1RandomBase) = <Base of the ram location where the the RNG
is stored>;
PUKCL _ZpEccRandomiseCoordinate(nu1PointBase) = <Base of the ram location of the point>;
PUKCL _ZpEccRandomiseCoordinate(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEccRandomiseCoordinate,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.6.9.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1RandomBase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1RandomBase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1RandomBase, u2ModLength + 4} and {nu1Workspace, 3*u2ModLength + 28}

#### 43.3.6.9.7. Status Returned Values

**Table 43-82.** ZpEccRandomiseCoordinate Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.6.10. Point is on Elliptic Curve

#### 43.3.6.10.1. Purpose

This service is used to test whether or not the point is on the curve.

#### 43.3.6.10.2. How to Use the Service

#### 43.3.6.10.3. Description

The operation performed is:

*Status* = *IsPointOnCurve*(*X*, *Y*, *Z*)

In this computation, the following parameters need to be provided:

- The input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointBase,3*u2ModLength + 4}). This Point can be the point at infinity.
- AParam and BParam are the Elliptic Curve Equation parameters. (pointed by{nu1AParam, u2ModLength+4} and {nu1BParam, u2ModLength+4}).
- Cns the Fast Modular Constant filled (pointed by{nu1CnsBase,u2ModLength+8}).
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4}).
- The workspace not initialized (pointed by {nu1WorkSpace, 4*u2ModLength +28}.

The result is the status of the point (X,Y,Z) regarding the Elliptic Curve Equation.

The service name for this operation is `ZpEcPointIsOnCurve`.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reduction service.

### 43.3.6.10.4. Parameters Definition

**Table 43-83.** ZpEcPointIsOnCurve Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointBase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point | unchanged |
| nu1AParam | nu1 | I | Crypto RAM | u2ModLength + 4 | The parameter a | The parameter a |
| nu1BParam | nu1 | I | Crypto RAM | u2ModLength + 4 | The parameter b | The parameter b |
| nu1Workspace | nu1 | I | Crypto RAM | 4*u2ModLength + 28 | – | Workspace |

### 43.3.6.10.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;


PUKCL (u2Option) = 0;

PUKCL _ZpEcPointIsOnCurve(nu1ModBase) = <Base of the ram location of P>;
PUKCL _ZpEcPointIsOnCurve(u2ModLength) = <Byte length of P>;
PUKCL _ZpEcPointIsOnCurve(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEcPointIsOnCurve(nu1AParam) = <Base of the ram location of the parameter a>;
PUKCL _ZpEcPointIsOnCurve(nu1BParam) = <Base of the ram location of the parameter b>;
PUKCL _ZpEcPointIsOnCurve(nu1PointBase) = <Base of the ram location of the point>;
PUKCL _ZpEcPointIsOnCurve(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEcPointIsOnCurve,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

### 43.3.6.10.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1AParam, nu1BParam, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength+4}, {nu1CnsBase, u2ModLength+8}, {nu1PointABase, 3*u2ModLength +12}, {nu1AParam, u2ModLength + 4}, {nu1BParam, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM.
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length.
- All overlapping between {nu1ModBase, u2ModLength+4}, {nu1CnsBase,u2ModLength+8}, {nu1PointABase, 3*u2ModLength+12}, {nu1AParam, u2ModLength+4}, {nu1AParam, u2ModLength + 4} and {nu1Workspace, 4*u2ModLength+28}.

### 43.3.6.10.7. Status Returned Values

**Table 43-84.** ZpEcPointIsOnCurve Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The point is on the curve. |
| PUKCL_POINT_IS_NOT_ON_ CURVE | Warning | The point is not on the curve. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-84.** ZpEcPointIsOnCurve Service Return Codes (continued)

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_POINT_AT_INFINITY | Warning | The input point has its Z equal to zero, so it's a representation of the infinite point. |

### 43.3.6.11. Generating an ECDSA Signature (Compliant with FIPS 186-2)

#### 43.3.6.11.1. Purpose

This service is used to generate an ECDSA signature following the FIPS 186-2. It performs the second step of the Signature Generation. A hash value *(HashVal)* must be provided as input, it has to be previously computed from the message to be signed using a secure hash algorithm.

A scalar number must be provided too as described in the FIPS 186-2. The result (R,S) is computed by this service.

#### 43.3.6.11.2. How to Use the Service

#### 43.3.6.11.3. Description

The operation performed is:

(*R*, *S*) = *EcDsaSign*($Pt_A$, *HashVal*, *k*, *CurveParameters*, *PrivateKey*)

This service processes the following checks:

- If the Scalar Number k is out of the range [1, PointOrder -1], the calculus is stopped and the status is set to PUKCL_WRONG_SELECT_NUMBER.
- If R equals zero, the calculus is stopped and the status is set to PUKCL_WRONG_SELECT_NUMBER.
- If S equals zero, the calculus is stopped and the status is set to PUKCL_WRONG_SELECT_NUMBER.

In this computation, the following parameters need to be provided:

- A the input point is filled in "mixed" coordinates (X,Y) with the affine values and Z = 1 (pointed by {nu1PointABase,3*u2ModLength + 12})
- Cns the working space for the Fast Modular Constant not initialized (pointed by {nu1CnsBase,u2ScalarLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})
- The workspace not initialized (pointed by {nu1WorkSpace, 8*u2ModLength + 44}
- The a parameter relative to the elliptic curve (pointed by {nu1ABase, u2ModLength + 4})
- The order of the Point A on the elliptic curve (pointed by {nu1OrderPointBase, u2ScalarLength + 4})
- k the input Scalar Number beforehand generated and filled (pointed by{nu1ScalarNumber,u2ScalarLength + 4})
- HashVal the hash value beforehand generated and filled (pointed by {nu1HashBase, u2ScalarLength + 4})
- The Private Key (pointed by {nu1PrivateKey, u2ScalarLength +4})
- Generally, u2ScalarLength is equal to (u2ModLength) or (u2ModLength + 4)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Important:**
For the ECDSA signature generation be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.

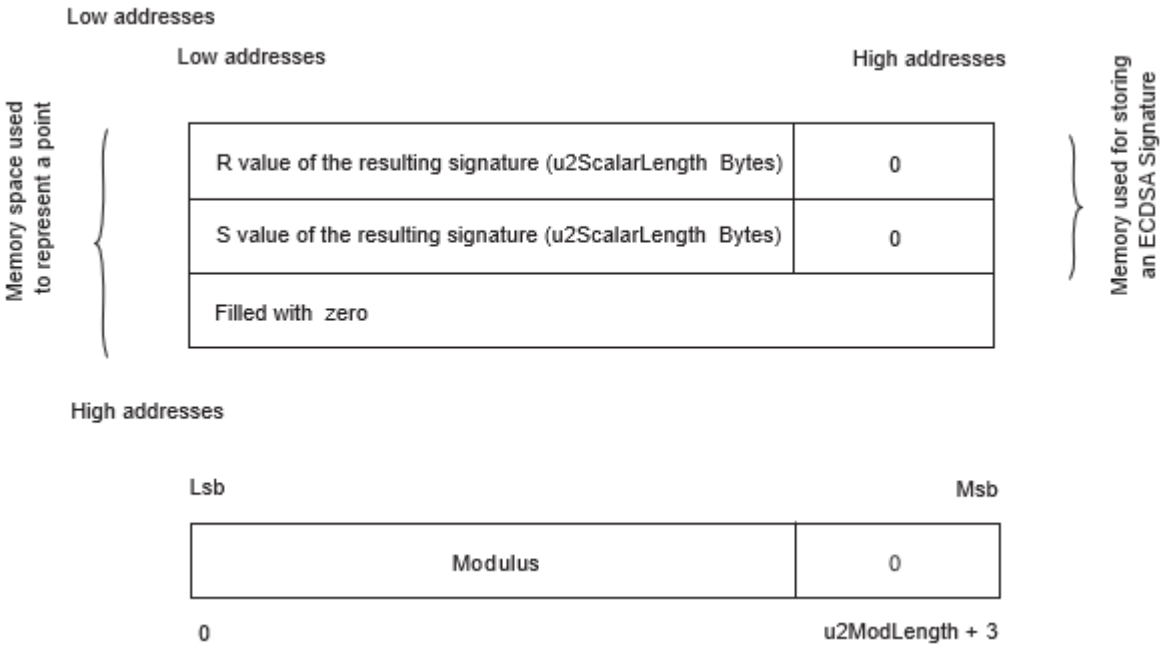
The scalar number k must be selected at random. This random must be generated before the call of the ECDSA signature. For this random generation be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.

The operation performed is:

- Compute the ECDSA (R,S) as described in FIPS 186-2, but leaving the user the role of computing the input Hash Value, thus leaving the freedom of using any other algorithm than SHA-1.
- Compute a R value using the input A point and the scalar number.
- Compute a S value using R, the scalar number, the private key and the provided hash value. Note that the resulting signature (R,S) is stored at the place of the input A point.
- If all is correct and S is different from zero, the status is set to PUKCL_OK. If all is correct and S equals zero,the status is set to PUKCL_WRONG_SELECT_NUMBER. If an error occurs, the status is set to the corresponding error value (see Status Returned Values below).

The service name for this operation is `ZpEcDsaGenerateFast`. This service uses Fast mode and Fast Modular Reduction for computation.

- The signature (R,S), when resulting from a computation is given back at address of the A point:
  - R output is at offset 0 and has length (u2ScalarLength + 4)bytes.
  - S output is at offset (u2ScalarLength + 4) bytes and has length (u2ScalarLength + 4) bytes.
  - The MSB 4 zero bytes may be suppressed to get the R and S values on u2ScalarLength bytes





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 43.3.6.11.4. Parameters Definition

**Table 43-85.** ZpEcDsaGenerateFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ScalarLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1ScalarNumber | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Scalar Number used to multiply the point A | Unchanged |
| nu1OrderPointBase | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Order of the Point A in the elliptic curve | Unchanged |
| nu1PrivateKey | nu1 | I/O | Crypto RAM | u2ScalarLength + 4 | Base of the Private Key | Unchanged |
| nu1HashBase (see **Note 1**) | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Base of the hash value resulting from the previous SHA | Unchanged |
| u2ScalarLength | u2 | I | – | – | Length of scalar (same length as the length of order) | Length of scalar |
| nu1PointABase (see **Note 2**) | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (three coordinates (X,Y) affine and Z = 1) | Resulting signature (R,S,0) |
| nu1ABase | nu1 | I | Crypto RAM | u2ModLength + 4 | Parameter a of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 8*u2ModLength + 44 | – | Corrupted workspace |

**Notes:**

1. The hash value calculus is defined by the ECDSA norm and depends on the elliptic curve domain parameters. To construct the input parameter, the 4 Most Significant Bytes must be set to zero.
2. The resulting signature format is different from the point A format (see Description above for information on the point A format).

### 43.3.6.11.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip


PUKCL (u2Option) = 0;

// Depending on the option specified, not all fields should be filled
PUKCL _ZpEcDsaGenerate(nu1ModBase) = <Base of the ram location of P>; PUKCL
_ZpEcDsaGenerate(u2ModLength) = <Byte length of P>;
PUKCL _ZpEcDsaGenerate(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _ZpEcDsaGenerate(nu1PointABase) = <Base of the A point>;
PUKCL _ZpEcDsaGenerate(nu1PrivateKey) = <Base of the Private Key>;
PUKCL _ZpEcDsaGenerate(nu1ScalarNumber) = <Base of the ScalarNumber>;
PUKCL _ZpEcDsaGenerate(nu1OrderPointBase) = <Base of the order of A point>;
PUKCL _ZpEcDsaGenerate(nu1ABase) = <Base of the a parameter of the curve>;
PUKCL _ZpEcDsaGenerate(nu1Workspace) = <Base of the workspace>;
PUKCL _ZpEcDsaGenerate(nu1HashBase) = <Base of the SHA resulting hash>;
PUKCL_ZpEcDsaGenerate(u2ScalarLength)    = < Length of ScalarNumber>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEcDsaGenerateFast, pvPUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
                {
                ...
                }
else // Manage the error
```

#### 43.3.6.11.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1PrivateKey, nu1ScalarNumber, nu1OrderPointBase,nu1ABase, nu1Workspace or nu1HashBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12},{nu1PrivateKey, u2ScalarLength + 4},{nu1ScalarNumber, u2ScalarLength + 4},{nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} or {nu1HashBase, u2ScalarLength + 4} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PrivateKey, u2ScalarLength + 4}, {nu1ScalarNumber, u2ScalarLength + 4}, {nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} and {nu1HashBase, u2ScalarLength + 4}

#### 43.3.6.11.7. Status Returned Values

**Table 43-86.** ZpEcDsaGenerateFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. The signature is the good one. |
| PUKCL_WRONG_SELECTNUMBER | Warning | The given value for nu1ScalarNumber is not good to perform this signature generation. |

### 43.3.6.12. Verifying an ECDSA Signature (Compliant with FIPS186-2)

#### 43.3.6.12.1. Purpose

This service is used to verify an ECDSA signature following the FIPS 186-2. It performs the second step of the Signature Verification.

A hash value (HashVal) must be provided as input, it has to be previously computed from the message to be signed using a secure hash algorithm.

As second significant input, the Signature is provided to be checked. This service checks the signature and fills the status accordingly.

#### 43.3.6.12.2. How to Use the Service

#### 43.3.6.12.3. Description

The operation performed is:

*Verify* = *EcDsaVerifySignature*($Pt_A$, *HashVal*, *Signature*, *CurveParameters*, *PublicKey*)

The points used for this operation are represented in different coordinate systems. In this computation, the following parameters need to be provided:

- A the input point is filled with the affine values (X,Y) and Z = 1 (pointed by{nu1PointABase,3*u2ModLength + 12})
- Cns the working space for the Fast Modular Constant not initialized (pointed by {nu1CnsBase,u2ScalarLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})
- The workspace not initialized (pointed by {nu1WorkSpace, 8*u2ModLength + 44}


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The a parameter relative to the elliptic curve (pointed by {nu1ABase,u2ModLength + 4})
- The order of the Point A on the elliptic curve (pointed by {nu1OrderPointBase,u2ScalarLength + 4})
- HashVal the hash value is generated prior and filled (pointed by {nu1HashBase,u2ScalarLength + 4})
- The Public Key point is filled in “mixed” coordinates (X,Y) with the affine values and Z = 1 (pointed by {nu1PointPublicKeyGen, 3*u2ModLength + 12})
- The input signature (R,S), even if it is not a Point, is represented in memory like a point in affine coordinates (X,Y) (pointed by {nu1PointSignature, 2*u2ScalarLength + 8})
  **Note:** For the ECDSA signature verification be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.
- The operation consists in obtaining a V value with all these input parameters and checking that V equals the provided R. If all is correct and the signature is the good one, the status is set to PUKCL_OK. If all is correct and the signature is wrong, the status is set to PUKCL_WRONG_SIGNATURE. If an error occurs, the status is set to the corresponding error value (see Status Returned Values below).

### 43.3.6.12.4. Parameters Definition

**Table 43-87.** ZpEcDsaVerifyFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ScalarLength + 12 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1OrderPointBase | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Order of the Point A in the elliptic curve | Unchanged |
| nu1PointSignature | nu1 | I | Crypto RAM | 2*u2ScalarLength + 8 | Signature(r, s) | Corrupted |
| nu1HashBase (see **Note 1**) | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Base of the hash value resulting from the previous SHA | Corrupted |
| u2ScalarLength | u2 | I | – | – | Length of scalar | Length of scalar |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Generator point | Corrupted |
| nu1PointPublicKeyGen | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Public point | Corrupted |
| nu1ABase | nu1 | I | Crypto RAM | u2ModLength + 4 | Parameter a of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 8*u2ModLength + 44 | – | Corrupted workspace |

**Note:**

1. The hash value calculus is defined by the ECDSA norm and depends on the elliptic curve domain parameters. To construct the input parameter, the 4 Most Significant Bytes must be set to zero.

### 43.3.6.12.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL(u2Option) = 0;

// Depending on the option specified, not all fields should be filled
PUKCL_ZpEcDsaVerify(nu1ModBase) = <Base of the ram location of P>;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL_ZpEcDsaVerify(u2ModLength) = <Byte length of P>;
PUKCL_ZpEcDsaVerify(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL_ZpEcDsaVerify(nu1PointABase) = <Base of the A point>;
PUKCL_ZpEcDsaVerify(nu1PrivateKey) = <Base of the Private Key>;
PUKCL_ZpEcDsaVerify(nu1ScalarNumber) = <Base of the ScalarNumber>;
PUKCL_ZpEcDsaVerify(nu1OrderPointBase) = <Base of the order of A point>;
PUKCL_ZpEcDsaVerify(nu1ABase) = <Base of the a parameter of the curve>;
PUKCL_ZpEcDsaVerify(nu1Workspace) = <Base of the workspace>;
PUKCL_ZpEcDsaVerify(nu1HashBase) = <Base of the SHA resulting hash>;
 PUKCL_ZpEcDsaVerify(u2ScalarLength)    = < Length of ScalarNumber>;

...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEcDsaVerifyFast, pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
            {
            ...
            }ou
else
            if(PUKCL(u2Status) == PUKCL_WRONG_SIGNATURE)
                        {
                        ...
                        }
            else // Manage the error
```

#### 43.3.6.12.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1PointPublicKeyGen, nu1PointSignature, nu1OrderPointBase,nu1ABase, nu1Workspace or nu1HashBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1PointPublicKeyGen, 3*u2ModLength + 12}, {nu1PointSignature,2*u2ScalarLength + 8}, {nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} or {nu1HashBase, u2ScalarLength + 4} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PointPublicKeyGen, 3*u2ModLength + 12}, {nu1PointSignature, 2*u2ScalarLength + 8}, {nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} and {nu1HashBase, u2ScalarLength + 4}

#### 43.3.6.12.7. Status Returned Values

**Table 43-88.** ZpEcDsaVerifyFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. The signature is the good one. |
| PUKCL_WRONG_SIGNATURE | Warning | The signature is wrong. |

### 43.3.6.13. Quick Verifying an ECDSA Signature (Compliant with FIPS 186-2)

#### 43.3.6.13.1. Purpose

This service is used to verify an ECDSA signature following the FIPS 186-2. It performs the second step of the Signature Verification using Quick Dual Multiplying to perform computation.

A hash value (HashVal) must be provided as input, it has to be previously computed from the message whose signature is verified using a secure hash algorithm.

As second significant input, the Signature is provided to be checked.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This service checks the signature and fills the status accordingly.

**Important:** This service has a quick implementation without additional security.

### 43.3.6.13.2. How to Use the Service

### 43.3.6.13.3. Description

The operation performed is:

*Verify* = *EcDsaVerifySignature*($Pt_A$, *HashVal*, *Signature*, *CurveParameters*, *PublicKey*)

The points used for this operation are represented in different coordinate systems.

In this computation, the following parameters need to be provided (such that u2MaxLength = max(u2ModLength, u2ScalarLength)):

- A the input point is filled with the affine values (X,Y) and Z = 1 (pointed by {pu1PointABase, (3*(u2ModLength + 4)) * ($2^{(WA-2)}$)})
- P the modulus filled and Cns the working space for the Fast Modular Constant not initialized (pointed by {pu1ModBase, u2ModLength + u2MaxLength + 16})
- The a parameter relative to the elliptic curve filled and workspace not initialized (pointed by {pu1AWorkBase,8*u2MaxLength + u2ModLength + 48})
- The order of the Point A on the elliptic curve (pointed by {pu1OrderPointBase,u2ScalarLength +4})
- HashVal the hash value beforehand generated and filled (pointed by {pu1HashBase,u2MaxLength +4})
- The Public Key point is filled in “mixed” coordinates (X,Y) with the affine values and Z = 1 (pointed by {nu1PointPublicKeyGen, (3*(u2ModLength + 4)) * ($2^{(WB-2)}$)})
- The input signature (R,S), even if it is not a Point, is represented in memory like a point in affine coordinates (X,Y) (pointed by {nu1PointSignature, 2*u2ScalarLength + 8})

The operation consists of obtaining a V value with all input parameters and checks that V equals the provided R. If all is correct and the signature is the good one, the status is set to PUKCL_OK. If all is correct and the signature is wrong, the status is set to PUKCL_WRONG_SIGNATURE. If an error occurs, the status is set to the corresponding error value (see Status Returned Values below).

### 43.3.6.13.4. Parameters Definition

To place the parameters correctly the maximum of u2ModLength and u2ScalarLength must be calculated: u2MaxLength = max(u2ModLength, u2ScalarLength)

WA is the Point A window size and WB is the Point Public Key window size (see Options below for details).

**Important:** Please calculate precisely the length of areas with the formulas and the `max()` service which takes the maximum of two values. Ensure that the pu1 type is a pointer on 4 bytes and contains the full address (see Aligned Significant Length for details).

**Table 43-89.** ZpEcDsaQuickVerify Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| pu1ModCnsBase | pu1 | I | Crypto RAM | u2ModLength + 4 + u2MaxLength + 12 | Base of modulus P | Base of modulus P |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-89.** ZpEcDsaQuickVerify Service Parameters (continued)

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| u2Option | u2 | I | – | – | Option related to the called service (see below) | – |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| pu1OrderPointBase | pu1 | I | Crypto RAM | u2ScalarLength + 4 | Order of the Point A in the elliptic curve | Unchanged |
| pu1PointSignature | pu1 | I | Any RAM | 2*u2ScalarLength + 8 | Signature(r, s) | Corrupted |
| pu1HashBase (see **Note 1**) | pu1 | I | Crypto RAM | u2MaxLength + 4 | Base of the hash value resulting from the previous SHA | Corrupted |
| u2ScalarLength | u2 | I | – | – | Length of scalar | Length of scalar |
| pu1PointABase | pu1 | I/O | Crypto RAM | (3*u2ModLength + 12) * (2(WA-2)) | Generator point | Corrupted |
| pu1PointPublicKeyGen | pu1 | I/O | Crypto RAM | (3*u2ModLength + 12) * (2(WB-2)) | Public Key point | Corrupted |
| pu1AWorkBase | pu1 | I | Crypto RAM | (u2ModLength + 4) + (8*u2MaxLength + 44) | Parameter a of the elliptic curve and Workspace | Corrupted |

**Note:**

1. 1. The hash value calculus is defined by the ECDSA norm and depends on the elliptic curve domain parameters. To construct the input parameter, the 4 Most Significant Bytes must be set to zero.

A suggested parameters placement in Crypto RAM is:

- ModCnsBase
- OrderPointBase
- Signature may be placed here or in Classical RAM
- HashBase
- PointABase
- PointPublicKeyGen
- AWorkBase

### 43.3.6.13.5. Options

The options are set by the u2Options input parameter, which is composed of:

- the mandatory windows sizes WA (window for Point A) and WB (window for Point Public Key)
- the indication of the presence of the Point Signature in system RAM

**Important:** Please check precisely if the Point Signature is in Crypto RAM. If this is the case the PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM must not be used.

The u2Options number is calculated by an “Inclusive OR” of the options. Some Examples in C language are:

- ```
  // Point Signature in system RAM
  // The Point A window size is 3
  // The Point Public Key window size is 4
  ```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.
  ```

```
PUKCL(u2Options) = PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM |
PUKCL_ZPECCMUL_WINSIZE_A_VAL_TO_OPT(3) |
PUKCL_ZPECCMUL_WINSIZE_B_VAL_TO_OPT(4);
```

- ```
  // Point Signature in the Cryptographic RAM
  // The Point A window size is 2
  // The Point Public Key window size is 5
  PUKCL(u2Options) = PUKCL_ZPECCMUL_WINSIZE_A_VAL_TO_OPT(2) |
  PUKCL_ZPECCMUL_WINSIZE_B_VAL_TO_OPT(5);
  ```

For this service, many window sizes are possible. The window sizes in bits are those of the windowing method used for the scalar multiplying.

The choice of the window sizes is a balance between the size of the parameters and the computation time:

- Increasing the window size increases the precomputation table size.
- Increasing the window size to the optimum reduces the computation time.

The following table details the estimated windows WA and WB optimum and possible for some curves.

**Table 43-90.** ZpEcDsaQuickVerify Service Estimated WA and WB Window Size

| Curve Size (bits) | Optimum Window size | Possible Window Sizes (WA, WB) or (WB, WA) |
|---|---|---|
| 192 | 5 | 5, 5 |
| 256 | 5 | 5, 5 |
| 384 | 6 | 5, 5 |
| 521 | 6 | 4, 5 |

The following table details the size of the point and the precomputation table, depending on the chosen window size option.

**Table 43-91.** ZpEcDsaQuickVerify Service Window Size and Precomputation Table Size Options

| Option Specified | Point and Precomputation Table Size |
|---|---|
| PUKCL_ZPECCMUL_WINSIZE_A_VAL_TO_OPT(WA) WA in [2, 15] | (3*(u2ModLength + 4)) * ($2^{(WA-2)}$) |
| PUKCL_ZPECCMUL_WINSIZE_B_VAL_TO_OPT(WB) WB in [2, 15] | (3*(u2ModLength + 4)) * ($2^{(WB-2)}$) |

The Point Signature can be located in PUKCC RAM or in system RAM. If the Point Signature is entirely in system RAM with no part in PUKCC RAM this can be signaled by us ing the option PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM. In all other cases this option must not be used.

The following table describes this option.

**Table 43-92.** ZpEcDsaQuickVerify Service Point Signature in Classical RAM Option

| Option | Purpose |
|---|---|
| PUKCL_ZPECCMUL_SCAL_IN_CLASSIC_RAM | The Point Signature can be located in Crypto RAM or in system RAM. If the Point Signature is entirely in system RAM with no part in PUKCC RAM this can be signaled by using this option. In all other cases this option must not be used. |

### 43.3.6.13.6. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;
PUKCL(u2Option) = <Point Signature location and windows sizes>;
PUKCL_ZpEcDsaQuickVerify(pu1ModCnsBase) = <Base of the ram location of P and Cns>;
PUKCL_ZpEcDsaQuickVerify(u2ModLength) = <Byte length of P>;
PUKCL_ZpEcDsaQuickVerify(pu1PointABase) = <Base of the ram location of the A point>;
PUKCL_ZpEcDsaQuickVerify(pu1PointPublicKeyGen) = <Base of the Public Key>;
PUKCL_ZpEcDsaQuickVerify(pu1PointSignature) = <Base of the Signature (r, s)>;
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.
```

```
PUKCL_ZpEcDsaQuickVerify(pu1OrderPointBase) = <Base of the order of the A point>;
PUKCL_ZpEcDsaQuickVerify(pu1AWorkBase) = <Base of the ram location of the parameter A of the
elliptic curve and workspace>;
PUKCL_ZpEcDsaQuickVerify(pu1HashBase) = <Base of the SHA resulting hash>;
PUKCL_ZpEcDsaQuickVerify(u2ScalarLength) = <Byte length of R and S in Point Signature>;
. . .
// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(ZpEcDsaQuickVerify, pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
                {
                ...
                }
else
                if ( PUKCL(u2Status) = PUKCL_WRONG_SIGNATURE )
                                {
                                ...
                                }
                else // Manage the error
```

#### 43.3.6.13.7. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- pu1ModCnsBase, pu1PointABase, pu1PointPublicKeyGen, pu1PointSignature,pu1OrderPointBase, pu1AWorkBase or pu1HashBase are not aligned on 32-bit boundaries
- {pu1ModCnsBase, u2ModLength + 4 + u2MaxLength + 12}, {pu1PointABase, (3 * u2ModLength + 12)* ($2^{(WA-2)}$)}, {pu1PointPublicKeyGen, (3 * u2ModLength + 12) * ($2^{(WPub-2)}$)}, {pu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {pu1AWorkBase, (u2ModLength + 4) + (8 * u2MaxLength + 44)} or {nu1HashBase, u2ScalarLength + 4} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {pu1ModCnsBase, u2ModLength + 4 + u2MaxLength + 12}, {pu1PointABase, (3 * u2ModLength + 12) * ($2^{(WA-2)}$)}, {pu1PointPublicKeyGen, (3 * u2ModLength + 12) *($2^{(WPub-2)}$)}, {pu1OrderPointBase, u2ScalarLength + 4}, {pu1PointSignature, 2 * u2ScalarLength + 8}, {nu1ABase, u2ModLength + 4}, {pu1AWorkBase, (u2ModLength + 4) + (8 * u2MaxLength + 44)} and {nu1HashBase, u2ScalarLength + 4}

#### 43.3.6.13.8. Status Returned Values

**Table 43-93.** ZpEcDsaQuickVerify Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. The signature is the good one. |
| PUKCL_WRONG_SIGNATURE | Warning | The signature is wrong. |

#### 43.3.6.13.9. Parameter Placement

The parameters' placement is described in detail in the following figures.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 43-11.** Modulus P and Cns{pu1ModCnsBase, u2ModLength + 4 + u2MaxLength + 12}

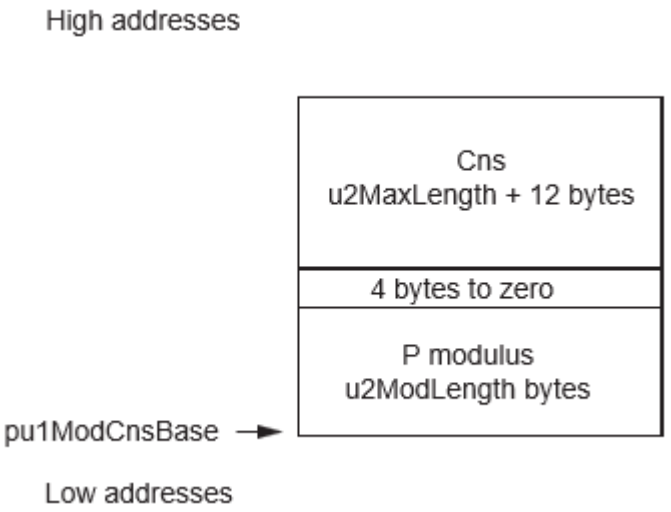
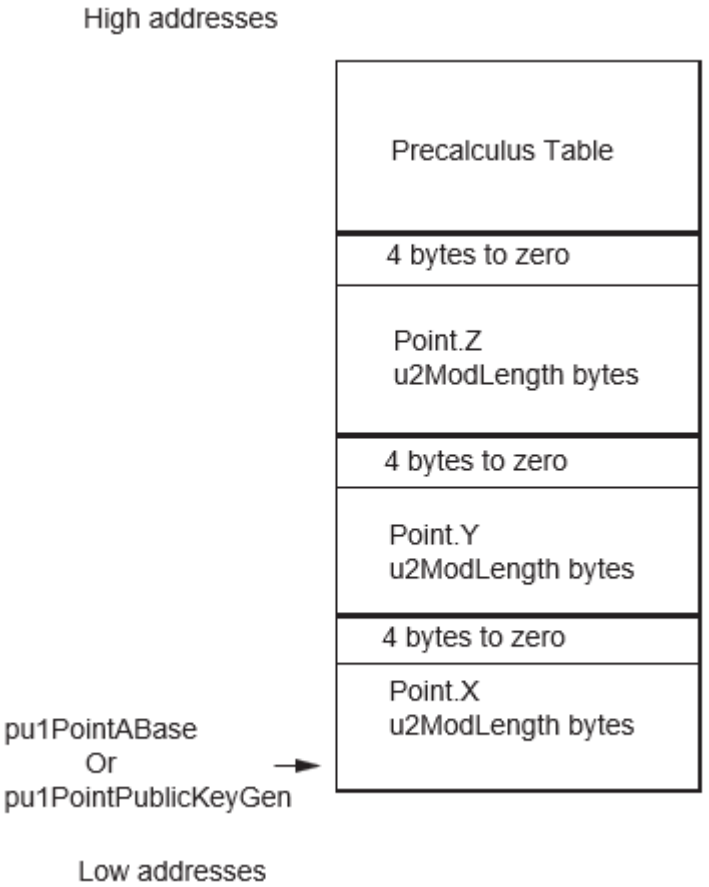



**Figure 43-12.** Points A {pu1PointABase, (3*(u2ModLength + 4)) * ($2^{(WA-2)}$)} and Public Key Gen {pu1PointPublicKeyGen, (3*(u2ModLength + 4)) * ($2^{(WB-2)}$)}





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 43-13.** PointSignature {pu1PointSignature, 2 * u2ScalarLength + 8}

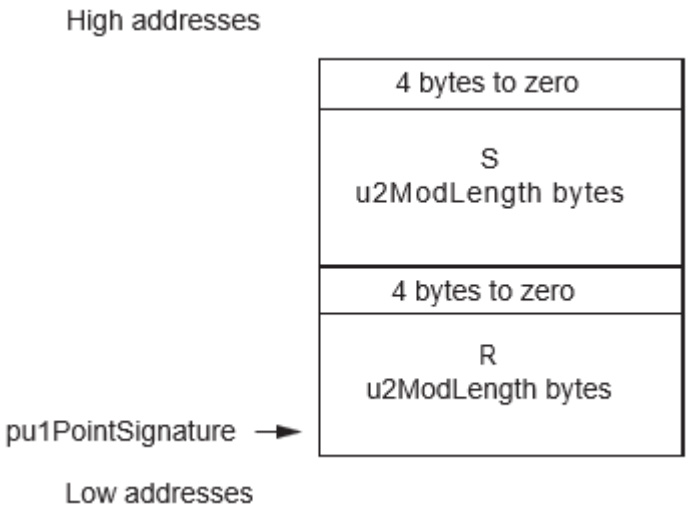
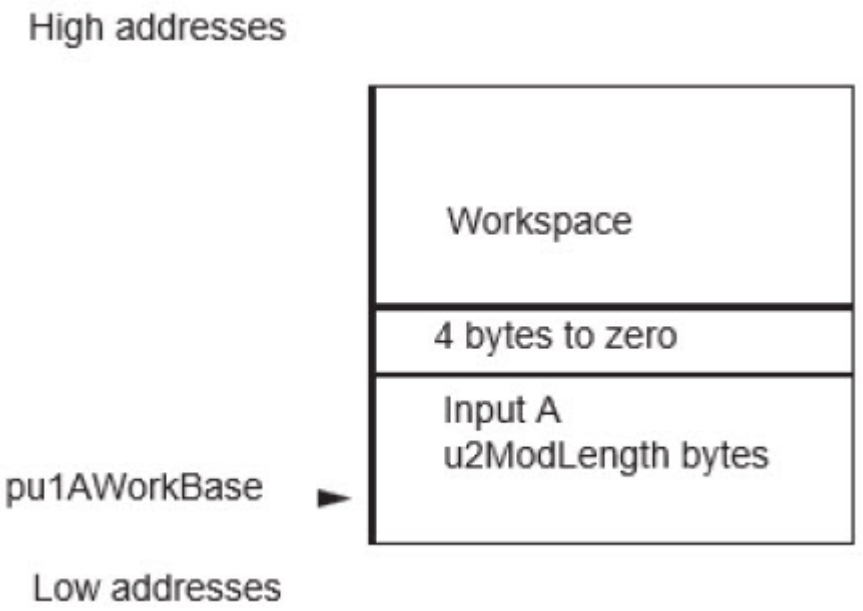



**Figure 43-14.** The a parameter and Workspace {pu1AWorkBase, 9*u2ModLength + 48}




### 43.3.7. Elliptic Curves Over GF($2^n$) Services

This section provides a complete description of the currently available elliptic curve over Polynomials in GF($2^n$) services.

These services process Polynomials in GF($2^n$) only.

The offered services cover the basic operations over elliptic curves such as:

- Adding two points over a curve
- Doubling a point over a curve
- Multiplying a point by an integral constant
- Converting a point's projective coordinates (resulting from a doubling or an addition) to the affine coordinates, and oppositely converting a point's affine coordinates to the projective coordinates.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Testing the point presence on the curve.

Additionally, some higher level services covering the needs for signature generation and verification are offered:

- Generating an ECDSA signature (compliant with FIPS186-2)
- Verifying an ECDSA signature (compliant with FIPS 186-2) The supported curves use the following curve equation in GF($2^n$):

$Y^2 + XY = X^3 + aX + b$

### 43.3.7.1. Parameters Format

#### 43.3.7.1.1. Polynomials in GF($2^n$)

Polynomials in GF($2^n$) are binary polynomials reduced modulo the polynomial P[X]. This polynomial is called the modulus and may be abbreviated to P in this document. The storage of these polynomials in memory area is described in Aligned Significant Length.

For notation simplicity the comparison signs "<" or ">" may be used for polynomials, this is to be interpreted as a comparison between the degree of the polynomials.

In GF($2^n$) fully reduced polynomials are of degree strictly lower than degree(P[X]). In many cases the polynomials used in this library are only partially reduced and so have a degree higher or equal than degree(P[X]), but this degree is maintained strictly lower than (degree(P[X]) + 15).

#### 43.3.7.1.2. Coordinates System

In this implementation, several choices have been made related to the coordinate systems managed by the elliptic curve primitives.

There are two systems currently managed by the library:

- Affine Coordinates System where each curve point has two coordinates (X,Y)
- Projective Coordinates System where each point is represented with three coordinates (X,Y,Z)

Converting from the affine coordinates system to a projective coordinates system and is performed by extending its representation having Z = 1:

$(X,Y) \Rightarrow (X,Y,Z=1)$

Converting from a projective coordinate to an affine one is a service offered by the library. The formula to perform this conversion is:

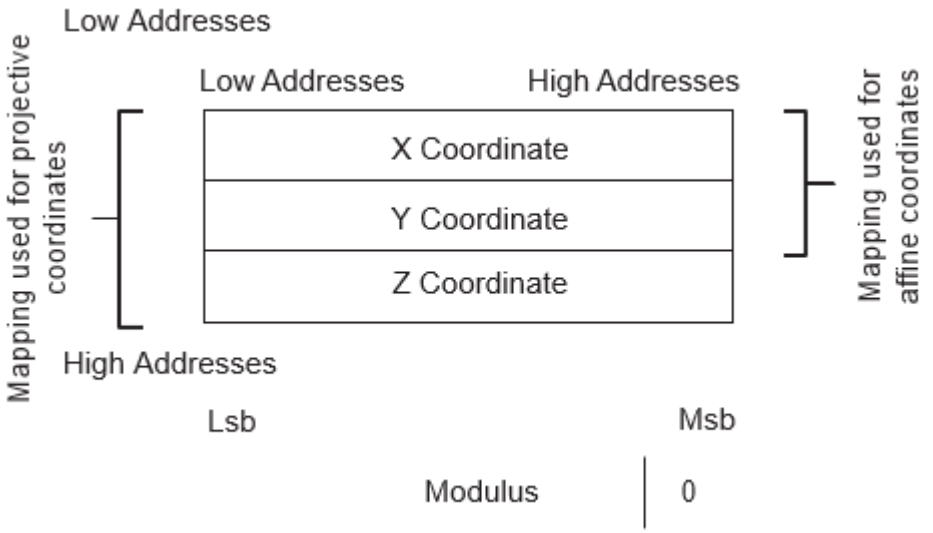
$(X,Y,Z) \Rightarrow (X \Rightarrow Z, Y/Z^2)$

#### 43.3.7.1.3. Points Representation in Memory

Depending on the representation (Projective or Affine), points are represented in memory as shown in the following figure.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 43-15.** Point Representation in Memory




In this figure, the modulus is represented as a reference, and to show that coordinates are always to be provided on the length of the modulus plus one 32-bit word.

Different types of representations are listed here:

Affine representation: $Pt = \begin{bmatrix} X_{Affine} < P \times X^{15} \\ Y_{Affine} < P \times X^{15} \end{bmatrix}$

Projective representation: $Pt = \begin{bmatrix} X_{\mathrm{Projective}} < P \times X^{15} \\ Y_{\mathrm{Projective}} < P \times X^{15} \\ Z_{\mathrm{Projective}} < P \times X^{15} \end{bmatrix}$

**Notes:**

1. The minimum value for u2ModLength is 12 bytes. Therefore, the significant length of the modulus must be at least three 32-bit words.
2. In some cases the point can be the infinite point. In this case it is represented with its Z coordinates equal or congruent to zero.

#### 43.3.7.1.4. Modulus and Modular Constant Parameters

In most of the services the following parameters must be provided:

- P the Modulus (often pointed by {nu1ModBase,u2ModLength + 4}): This parameter contains the Modulus Polynomial P[X] defining the Galois Field used in points coordinates computations. The Modulus must be u2ModLength bytes long, while having a supplemental zeroed 32-bit word on the MSB side.
  **Note:** Most of the Elliptic Curve computations are reduced modulo P. In many functions the reductions are made with the Fast Reduction.
- Cns the Modular Constant (often pointed by {nu1CnsBase,u2ModLength + 12}): This parameter contains the Modular Constant associated to the Modulus.

**Important:** The Modular Constant must be calculated before using the GF($2^n$) Elliptic Curves functions by a call to the Setup for Modular Reductions with the GF($2^n$) option (see Modular Reduction).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.7.1.5. Curve Parameters in Memory

Some services need one or both of the Elliptic Curve Equation Parameters a and b. In this case these values are organized in memory as follows:

- The a Parameter relative to the Elliptic Curve Equation (often pointed by {nu1ABase,u2ModLength +4}). The a Parameter is written in a classical way in memory. It is u2ModLength bytes long and has a supplemental zeroed 32-bit word on the MSB side.
- The a and b Parameters relative to the Elliptic Curve Equation (often pointed by {nu1ABBase,2*u2ModLength + 8}):
  - The a Parameter is written in memory on u2ModLength bytes long, with a supplemental zeroed 32-bit word on the MSB side.
  - The b Parameter is written in memory after the a Parameter at an offset of (u2ModLength + 4) bytes. It is written in memory on u2ModLength bytes long, with a supplemental zeroed 32-bit word on the MSB side.

### 43.3.7.2. Point Addition

#### 43.3.7.2.1. Purpose

This service is used to perform a point addition, based on a given elliptic curve over GF($2^n$).

Please note that this service is not intended to add the same point twice. In this particular case, use the doubling service (see Point Doubling).

#### 43.3.7.2.2. How to Use the Service

#### 43.3.7.2.3. Description

The operation performed is:

$Pt_C = Pt_A + Pt_B$

In this computation, the following parameters need to be provided:

- Point A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Point B the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointBBase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength + 12})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})
- The a parameter relative to the elliptic curve equation (pointed by {nu1ABase,u2ModLength + 4})
- The workspace not initialized (pointed by {nu1WorkSpace, 7*u2ModLength + 40}

The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the same place than the input point A. This Point can be the Infinite Point.

The services for this operation are:

- Service GF2NEccAddFast: The fast mode is used, the fast modular reduction is used in the computations.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Modular Reductions service.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.7.2.4. Parameters Definition

**Table 43-94.** GF2NEccAddFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of Modulus P | Base of Modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 12 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulo | Length of modulo |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (projective coordinates) | Resulting point C (projective coordinates) |
| nu1PointBBase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point B (projective coordinates) | Input point B |
| nu1ABBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Parameter a of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 7*u2ModLength + 40 | – | Corrupted workspace |

#### 43.3.7.2.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;
//Depending on the function the Random Number Generator
//must be initialized and started
//following the directives given for the RNG on the chip
PUKCL(u2Option) = 0;
PUKCL_GF2NEccAdd(nu1ModBase) = <Base of the ram location of P>;
PUKCL_GF2NEccAdd(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL_GF2NEccAdd(u2ModLength) = <Byte length of P>;
PUKCL_GF2NEccAdd(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL_GF2NEccAdd(nu1PointBBase) = <Base of the ram location of the B point>;
PUKCL_GF2NEccAdd(nu1ABBase) = <Base of the ram location of the a Parameter>;
PUKCL_GF2NEccAdd(nu1Workspace) = <Base of the ram location of the workspace>;
. . .
// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEccAddFast, pvPUKCLParam);
if (PUKCL(u2Status) == PUKCL_OK)
                {
                ...
                }
else // Manage the error
```

#### 43.3.7.2.6. Constraints

No overlapping between either input and output are allowed The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase,nu1CnsBase, nu1PointABase, nu1PointBBase, nu1ABBase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1PointBBase, 3*u2ModLength + 12}, {nu1ABase,u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PointBBase, 3*u2ModLength + 12}, {nu1ABase,u2ModLength + 4} and {nu1Workspace, 5*u2ModLength + 32}

#### 43.3.7.2.7. Status Returned Values

**Table 43-95.** GF2NEccAddFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without errors. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 43.3.7.3. Point Doubling

#### 43.3.7.3.1. Purpose

This service is used to perform a Point Doubling, based on a given elliptic curve over GF($2^n$).

#### 43.3.7.3.2. How to Use the Service

#### 43.3.7.3.3. Description

The operation performed is:

$Pt_C = 2 \times Pt_A$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength +8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 4*u2ModLength +28}
- The a and b Parameters relative to the Elliptic Curve Equation (pointed by {nu1ABBase,2*u2ModLength+ 8})
- The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the very same place than the input point A. This point can be the Infinite Point.

The service name for this operation is `GF2NEccDblFast`. This service uses Fast mode and Fast Modular Reduction for computation.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reductions service.

#### 43.3.7.3.4. Parameters Definition

**Table 43-96.** GF2NEccDblFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 12 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1ABBase | u2 | I | Crypto RAM | 2*u2ModLength + 8 | Parameters a and b of the elliptic curve | Parameter a and b of the elliptic curve |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (projective coordinates) | Resulting point C (projective coordinates) |
| nu1Workspace | nu1 | I | Crypto RAM | 4*u2ModLength + 28 | – | Corrupted workspace |

#### 43.3.7.3.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

PUKCL (u2Option) = 0;

PUKCL _GF2NEccDbl(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEccDbl(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEccDbl(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEccDbl(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _GF2NEccDbl(nu1ABBase) = <Base of the a and b parameters of the elliptic curve>;
PUKCL _GF2NEccDbl(nu1Workspace) = <Base of the ram location of the workspace>;
...
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEccDblFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.7.3.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1ABBase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12}, {nu1ABBase, 2*u2ModLength + 8}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1ABase, u2ModLength + 4} and {nu1Workspace, 4*u2ModLength + 28}

#### 43.3.7.3.7. Status Returned Values

**Table 43-97.** GF2NEccDblFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.7.4. Scalar Point Multiply

#### 43.3.7.4.1. Purpose

This service is used to multiply a point by an integral constant K on a given elliptic curve over GF($2^n$).

#### 43.3.7.4.2. How to Use the Service

#### 43.3.7.4.3. Description

The operation performed is:

$Pt_C = K \times Pt_A$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointABase,3*u2ModLength + 12}). This point can be the Infinite Point.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})
- The workspace not initialized (pointed by {nu1WorkSpace, 8*u2ModLength + 44}
- The a and b parameters relative to the elliptic curve (pointed by {nu1ABBase,2*u2ModLength + 8})
- K the scalar number (pointed by {nu1ScalarNumber,u2ScalarLength + 4})

The resulting C point is represented in projective coordinates (X,Y,Z) and is stored at the very same place than the input point A. This point can be the Infinite Point.

The service name for this operation is `GF2NEccMulFast`. This service uses Fast mode and Fast Modular Reduction for computation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reductions service.

#### 43.3.7.4.4. Parameters Definition

**Table 43-98.** GF2NEccMulFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 12 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1KBase | nu1 | I | Crypto RAM | u2KLength | Scalar number used to multiply the point A | Unchanged |
| u2KLength | u2 | I | – | – | Length of scalar K | Length of scalar K |
| nu1PointBase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (projective coordinates) | Resulting point C (projective coordinates) |
| nu1ABase | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Parameters a and b of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 8*u2ModLength + 44 | – | Corrupted workspace |

#### 43.3.7.4.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

PUKCL (u2Option) = 0;

PUKCL _GF2NEccMul(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEccMul(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEccMul(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEccMul(nu1PointBase) = <Base of the ram location of the A point>;
PUKCL _GF2NEccMul(nu1ABase) = <Base of the ram location of the parameters a and b of the
elliptic
curve>;
PUKCL _GF2NEccMul(nu1KBase) = <Base of the ram location of the scalar number>;
PUKCL _GF2NEccMul(nu1Workspace) = <Base of the ram location of the workspace>;
PUKCL _GF2NEccMul(u2KLength) = <Length of the ram location of the scalar number>;

...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEccMulFast,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.7.4.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointBase, nu1ABase, nu1KBase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointBase, 3*u2ModLength+ 12}, {nu1ABase, 2*u2ModLength + 8}, {nu1KBase, u2KLength} or {nu1Workspace, 8*u2ModLength + 44} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointBase, 3*u2ModLength + 12}, {nu1ABase, 2*u2ModLength + 8}, {nu1KBase, u2KLength} and {nu1Workspace, 8*u2ModLength + 44}

#### 43.3.7.4.7. Status Returned Values

**Table 43-99.** GF2NEccMulFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.7.5. Projective to Affine Coordinates Conversion

#### 43.3.7.5.1. Purpose

This service is used to perform a point coordinates conversion from a projective representation to an affine.

#### 43.3.7.5.2. How to Use the Service

#### 43.3.7.5.3. Description

The operation performed is:

$$Pt_{X\ Affine\ \ coordinate} = \left[\frac{Pt_{X\mathrm{Projective\ coordinate}}}{\left(Pt_{Z\ \mathrm{Projective\ coordinate}}\right)}\right]$$

$$Pt_{Y\ Affine\ \ coordinate} = \left[\frac{Pt_{Y\ \mathrm{Projective\ coordinate}}}{\left(Pt_{Z\ \mathrm{Projective\ coordinate}}\right)2}\right]$$

In this computation, the following parameters need to be provided:

- A the input point is filled in projective coordinates (X,Y,Z) or affine coordinates for X and Y, and setting Z to 1 (pointed by {nu1PointABase,3*u2ModLength + 12}). The Point A can be the point at infinity. In this case, the u2Status returned is PUKCL_POINT_AT_INFINITY.
- Cns the Modular Constant filled (pointed by {nu1CnsBase,u2ModLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})
- The workspace not initialized (pointed by {nu1WorkSpace, 4*u2ModLength + 48}

The result is the point A with its (X,Y) coordinates converted to affine, and the Z coordinate set to 1.

The service name for this operation is `GF2NEcConvProjToAffine`.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reductions service.

#### 43.3.7.5.4. Parameters Definition

**Table 43-100.** GF2NEcConvProjToAffine Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 12 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointABase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point A | Resulting point A in affine coordinates |
| nu1Workspace | nu1 | I | Crypto RAM | 4*u2ModLength + 48 | – | Workspace |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.7.5.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL (u2Option) = 0;

PUKCL _GF2NEcConvProjToAffine(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEcConvProjToAffine(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEcConvProjToAffine(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEcConvProjToAffine(nu1PointABase) = <Base of the ram location of the A point>;
PUKCL _GF2NEcConvProjToAffine(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEcConvProjToAffine,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.7.5.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8},{nu1PointABase, 3*u2ModLength + 12}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12} and {nu1Workspace, 4*u2ModLength + 48}

#### 43.3.7.5.7. Status Returned Values

**Table 43-101.** GF2NEcConvProjToAffine Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |
| PUKCL_POINT_AT_INFINITY | Warning | The input point has its Z equal to zero, so it is a representation of the infinite point. |

### 43.3.7.6. Affine to Projective Coordinates Conversion

#### 43.3.7.6.1. Purpose

This service is used to perform a point coordinates conversion from an affine point representation to projective.

#### 43.3.7.6.2. How to Use the Service

#### 43.3.7.6.3. Description

The operation performed is:

*affine*(*Xa*, *Ya*) → *projective*(*Xp*, *Yp*, *Zp*)

In this computation, the following parameters need to be provided:

- A the input point is filled in affine coordinates for X and Y, and setting Z to 1 (pointed by {nu1PointABase,3*u2ModLength + 4}).
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The workspace not initialized (pointed by {nu1WorkSpace, 2*u2ModLength +16} The result is the point A with its (X,Y,Z) projective coordinates.

The service name for this operation is `GF2NEcConvAffineToProjective`.

**Important:** Before using this service, ensure that the constant Cns has been calculated with the setup of the Fast Modular Reductions service.

### 43.3.7.6.4. Parameters Definition

**Table 43-102.** GF2NEcConvAffineToProjective Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointABase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point A | Resulting point A in affine coordinates |
| nu1Workspace | nu1 | I | Crypto RAM | 2*u2ModLength + 16 | – | Workspace |

### 43.3.7.6.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL (u2Option) = 0;

PUKCL _GF2NEcConvAffineToProjective(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEcConvAffineToProjective(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEcConvAffineToProjective(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEcConvAffineToProjective(nu1PointABase) = <Base of the ram location of the A
point>;
PUKCL _GF2NEcConvAffineToProjective(nu1Workspace) = <Base of the ram location of the
workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEcConvAffineToProjective,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

### 43.3.7.6.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, and {nu1Workspace, 2*u2ModLength + 16}


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.7.6.7. Status Returned Values

**Table 43-103.** GF2NEcConvAffineToProjective Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.7.7. Randomize Coordinate

#### 43.3.7.7.1. Purpose

This service is used to convert the Projective representation of a point to another Projective representation.

#### 43.3.7.7.2. How to Use the Service

#### 43.3.7.7.3. Description

The operation performed is:

$Projective(X_1, Y_1, Z_1) \rightarrow Projective(X_2, Y_2, Z_2)$

In this computation, the following parameters need to be provided:

- The input point is filled in projective coordinates (X,Y,Z) (pointed by {nu1PointBase,3*u2ModLength + 12}). This Point must not be the point at infinity.
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase,u2ModLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})
- The workspace not initialized (pointed by {nu1WorkSpace, 3*u2ModLength + 28}
- The random number (pointed by {nu1RandomBase, u2ModLength + 4}) The result is the point nu1PointBase with its (X,Y,Z) coordinates randomized. The service for this operation is GF2NEcRandomiseCoordinate.

**Important:**
Before using this service:

- Ensure that the constant Cns has been calculated with the Setup of the fast Modular Reductions service.
- Be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.

#### 43.3.7.7.4. Parameters Definition

**Table 43-104.** GF2NEcRandomiseCoordinate Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointBase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point | Resulting point |
| nu1RandomBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Random | Corrupted |
| nu1Workspace | nu1 | I | Crypto RAM | 3*u2ModLength + 28 | – | Workspace |

#### 43.3.7.7.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
// ! following the directives given for the RNG on the chip

PUKCL (u2Option) = 0;

// Depending on the option specified, not all fields should be filled
PUKCL _GF2NEcRandomiseCoordinate(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEcRandomiseCoordinate(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEcRandomiseCoordinate(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL_GF2NEcRandomiseCoordinate(nu1RandomBase) = <Base of the ram location where the the rng
is stored>;
PUKCL _GF2NEcRandomiseCoordinate(nu1PointBase) = <Base of the ram location of the point>;
PUKCL _GF2NEcRandomiseCoordinate(nu1Workspace) =
<Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEcRandomiseCoordinate,&PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.7.7.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1RandomBase, nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1RandomBase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1RandomBase, u2ModLength + 4} and {nu1Workspace, 3*u2ModLength + 28}

#### 43.3.7.7.7. Status Returned Values

**Table 43-105.** GF2NEcRandomiseCoordinate Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |

### 43.3.7.8. Point is on Elliptic Curve

#### 43.3.7.8.1. Purpose

This service is used to test whether the point is on the curve.

#### 43.3.7.8.2. How to Use the Service

#### 43.3.7.8.3. Description

The operation performed is:

Status = IsPointOnCurve(X, Y, Z);

In this computation, the following parameters need to be provided:

- The input points filled in projective coordinates (X, Y, Z) (pointed by {nu1PointBase, 3*U2ModLength + 4}). This point can be point at infinity.
- AParam and BParam are the Elliptic Curve Equation parameters (pointed by {nu1AParam, u2ModLength+ 4} and {nu1BParam, u2ModLength + 4}).
- Cns the Fast Modular Constant filled (pointed by {nu1CnsBase, u2ModLength + 8})


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- P the modulus filled (pointed by {nu1ModBase, u2ModLength + 8})
- The workspace not initialized (pointed by {nu1WorkSpace, 4*u2ModLength + 28})

The service name for this operation is `GF2NEcPointIsOnCurve`.

**Important:** Before using this service, the constant Cns must have been calculated with the Fast Modular Reduction service.

#### 43.3.7.8.4. Parameters Definition

**Table 43-106.** GF2NEcPointIsOnCurve Service Parameters

| Parameter | Type | Dir. | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ModLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1PointBase | nu1 | I | Crypto RAM | 3*u2ModLength + 12 | Input point | Unchanged |
| nu1AParam | nu1 | I | Crypto RAM | u2ModLength + 4 | The parameter a | Unchanged |
| nu1BParam | nu1 | I | Crypto RAM | u2ModLength + 4 | The parameter b | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 4*u2ModLength + 28 | N/A | Workspace |

#### 43.3.7.8.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL (u2Option) = 0;

// Depending on the option specified, not all fields should be filled
PUKCL _GF2NEcPointIsOnCurve(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEcPointIsOnCurve(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEcPointIsOnCurve(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEcPointIsOnCurve(nu1PointABase) = <Base of the A point>;
PUKCL _GF2NEcPointIsOnCurve(nu1AParam) = <Base of the ram location of the parameter a>;
PUKCL _GF2NEcPointIsOnCurve(nu1BParam) = <Base of the ram location of the parameter b>;
PUKCL _GF2NEcPointIsOnCurve(nu1PointBase) = <Base of the ram location of the point>;
PUKCL _GF2NEcPointIsOnCurve(nu1Workspace) = <Base of the ram location of the workspace>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKC L_Process(GF2NEcPointIsOnCurve,
pvPUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```

#### 43.3.7.8.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure that the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1AParam, nu1BParam and nu1Workspace are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1AParam, u2ModLength + 4}, {nu1BParam, u2ModLength + 4}, {nu1Workspace, 4*u2ModLength + 28} are not in Crypto RAM


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1AParam, u2ModLength + 4}, {nu1BParam, u2ModLength + 4} and {nu1Workspace, 4*u2ModLength + 28}

#### 43.3.7.8.7. Status Returned Values

**Table 43-107.** GF2NEcPointIsOnCurve Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The point is on the curve. |
| PUKCL_POINT_IS_NOT_ON_CURVE | Warning | The point is not on the curve. |
| PUKCL_POINT_AT_INFINITY | Warning | The input point has its Z equal to zero, so it's a representation of the infinite point. |

### 43.3.7.9. Generating an ECDSA Signature (Compliant with FIPS 186-2)

#### 43.3.7.9.1. Purpose

This service is used to generate an ECDSA signature following the FIPS 186-2. It performs the second step of the Signature Generation. A hash value (HashVal) must be provided as input, it has to be previously computed from the message to be signed using a secure hash algorithm.

A scalar number must be provided, as described in the FIPS 186-2.

The result (R,S) is computed by this service. If S equals zero, the status is set to PUKCL_WRONG_SELECT_NUMBER.

#### 43.3.7.9.2. How to Use the Service

#### 43.3.7.9.3. Description

The operation performed is:

(*R*, *S*) = *EcDsaSign*($Pt_A$, *HashVal*, *k*, *CurveParameters*, *PrivateKey*)

This service processes the following checks:

- If the Scalar Number k is out of the range [1, PointOrder -1], the calculus is stopped and the status is set to PUKCL_WRONG_SELECT_NUMBER.
- If R equals zero, the calculus is stopped and the status is set to PUKCL_WRONG_SELECT_NUMBER.
- If S equals zero, the calculus is stopped and the status is set to PUKCL_WRONG_SELECT_NUMBER. In this computation, the following parameters need to be provided:
- A the input point is filled in "mixed" coordinates (X,Y) with the affine values and Z = 1 (pointed by {nu1PointABase,3*u2ModLength + 12})
- Cns the working space for the Fast Modular Constant not initialized (pointed by {nu1CnsBase,u2ScalarLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength + 4})
- The workspace not initialized (pointed by {nu1WorkSpace, 8*u2ModLength + 44}
- The a and b parameters relative to the elliptic curve equation (pointed by {nu1ABBase, 2*u2ModLength + 8})
- The order of the Point A on the elliptic curve (pointed by {nu1OrderPointBase, u2ScalarLength + 4})
- k the input Scalar Number beforehand generated and filled (pointed by{nu1ScalarNumber,u2ScalarLength + 4})
- HashVal the hash value beforehand generated and filled (pointed by {nu1HashBase, u2ScalarLength +4})


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The Private Key (pointed by {nu1PrivateKey, u2ScalarLength +4})
- Generally u2ScalarLength is equal to (u2ModLength) or (u2ModLength + 4)

**Important:**
For the ECDSA signature generation be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.

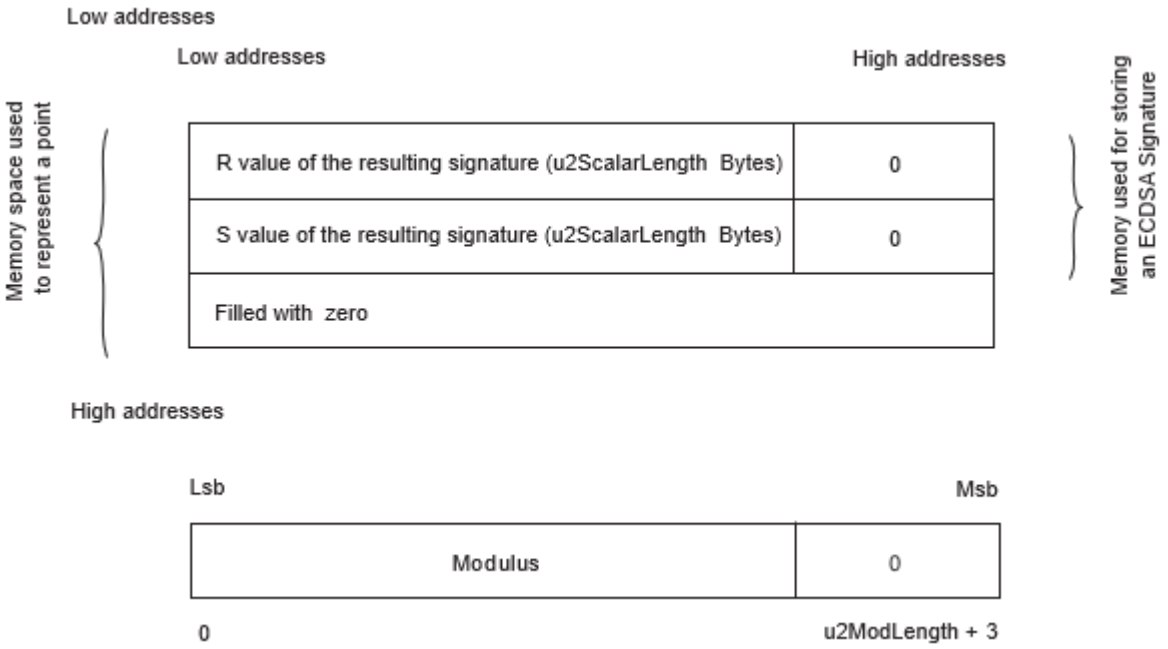
The scalar number k must be selected at random. This random must be generated before the call of the ECDSA signature. For this random generation be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.

The operation performed is:

- Compute the ECDSA (R,S) as described in FIPS 186-2, but leaving the user the role of computing the input Hash Value, thus leaving the freedom of using any other algorithm than SHA-1.
- Compute a R value using the input A point and the scalar number.
- Compute a S value using R, the scalar number, the private key and the provided hash value. Note that the resulting signature (R,S) is stored at the place of the input A point.
- If all is correct and S is different from zero, the status is set to PUKCL_OK. If all is correct and S equals zero,the status is set to PUKCL_WRONG_SELECT_NUMBER. If an error occurs, the status is set to the corresponding error value (see Status Returned Values below).

The service name for this operation is GF2NEcDsaGenerateFast. The fast mode is used, the fast modular reduction is used in the computations.

- The signature (R,S), when resulting from a computation is given back at address of the A point:
  - The R value result with u2ModLength + 4 bytes (padded with zeros).
  - The S value result with u2ModLength + 4 bytes (padded with zeros)
  - The u2NLength + 4 following bytes (space for the third coordinate of A) are filled with zeros.





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 43.3.7.9.4. Parameters Definition

**Table 43-108.** GF2NEcDsaGenerateFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ScalarLength + 12 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1ScalarNumber | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Scalar Number used to multiply the point A | Unchanged |
| nu1OrderPointBase | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Order of the Point A in the elliptic curve | Unchanged |
| nu1PrivateKey | nu1 | I/O | Crypto RAM | u2ScalarLength + 4 | Base of the Private Key | Unchanged |
| nu1HashBase (see **Note 1**) | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Base of the hash value resulting from the previous SHA | Unchanged |
| u2ScalarLength | u2 | I | – | – | Length of scalar (same length as the length of order) | Length of scalar |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Input point A (three coordinates (X,Y) affine and Z = 1) | Resulting signature (R,S,0) |
| nu1ABase | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Parameter a of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 8*u2ModLength + 44 | – | Corrupted workspace |

**Note:**

1. Whatever the chosen SHA, the resulting hash value may have a length inferior or equal to the modulo length and be padded with zeros until its total length is u2ModLength + 4.

### 43.3.7.9.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip

PUKCL (u2Option) = 0;

// Depending on the option specified, not all fields should be filled
PUKCL _GF2NEcDsaGenerate(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEcDsaGenerate(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEcDsaGenerate(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEcDsaGenerate(nu1PointABase) = <Base of the A point>;
PUKCL _GF2NEcDsaGenerate(nu1PrivateKey) = <Base of the Private Key>;
PUKCL _GF2NEcDsaGenerate(nu1ScalarNumber) = <Base of the ScalarNumber>;
PUKCL _GF2NEcDsaGenerate(nu1OrderPointBase) = <Base of the order of A point>;
PUKCL _GF2NEcDsaGenerate(nu1ABase) = <Base of the a parameter of the curve>; PUKCL
_GF2NEcDsaGenerate(nu1Workspace) = <Base of the workspace>;
PUKCL _GF2NEcDsaGenerate(nu1HashBase) = <Base of the SHA resulting hash>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEcDsaGenerateFast, pvPUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else // Manage the error
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 43.3.7.9.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1PrivateKey, nu1ScalarNumber, nu1OrderPointBase,nu1ABase, nu1Workspace or nu1HashBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength+ 12},{nu1PrivateKey, u2ScalarLength + 4},{nu1ScalarNumber, u2ScalarLength + 4},{nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} or {nu1HashBase, u2ScalarLength + 4} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PrivateKey, u2ScalarLength + 4}, {nu1ScalarNumber, u2ScalarLength + 4}, {nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABase, u2ModLength + 4}, {nu1Workspace, <WorkspaceLength>} and {nu1HashBase, u2ScalarLength + 4}

#### 43.3.7.9.7. Status Returned Values

**Table 43-109.** GF2NEcDsaGenerate Fast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without problem. |
| PUKCL_WRONG_SELECTNUMBER | Warning | The given value for nu1ScalarNumber is not good to perform this signature generation. |

### 43.3.7.10. Verifying an ECDSA Signature (Compliant with FIPS 186-2)

#### 43.3.7.10.1. Purpose

This service is used to verify an ECDSA signature following the FIPS 186-2. It performs the second step of the Signature Verification.

A hash value (HashVal) must be provided as input, it has to be previously computed from the message to be signed using a secure hash algorithm.

As second significant input, the Signature is provided to be checked. This service checks the signature and fills the status accordingly.

#### 43.3.7.10.2. How to Use the Service

#### 43.3.7.10.3. Description

The operation performed is:

*Verify* = *EcDsaVerifySignature*($Pt_A$, *HashVal*, *Signature*, *CurveParameters*, *PublicKey*)

The points used for this operation are represented in different coordinate systems. In this computation, the following parameters need to be provided:

- A the input point is filled with the affine values (X,Y) and Z = 1 (pointed by{nu1PointABase,3*u2ModLength + 12})
- Cns the working space for the Fast Modular Constant not initialized (pointed by {nu1CnsBase,u2ScalarLength + 8})
- P the modulus filled (pointed by {nu1ModBase,u2ModLength +4})
- The workspace not initialized (pointed by {nu1WorkSpace, 8*u2ModLength +44} The a and b parameters relative to the elliptic curve (pointed by {nu1ABase,2*u2ModLength + 8})
- The order of the Point A on the elliptic curve (pointed by {nu1OrderPointBase,u2ScalarLength +4})


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- HashVal the hash value beforehand generated and filled (pointed by {nu1HashBase,u2ScalarLength +4})
- The Public Key point is filled in “mixed” coordinates (X,Y) with the affine values and Z = 1 (pointed by {nu1PointPublicKeyGen, 3*u2ModLength + 12})
- The input signature (R,S), even if it is not a Point, is represented in memory like a point in affine coordinates (X,Y) (pointed by {nu1PointSignature, 2*u2ScalarLength + 8})

**Important:** For the ECDSA signature verification be sure to follow the directives given for the RNG on the chip you use (particularly initialization, seeding) and compulsorily start the RNG.

- The operation consists in obtaining a V value with all these input parameter and check that V equals the provided R. If all is correct and the signature is the good one, the status is set to PUKCL_OK. If all is correct and the signature is wrong, the status is set to PUKCL_WRONG_SIGNATURE. If an error occurs, the status is set to the corresponding error value (see Status Returned Values below).

The service name for this operation is `GF2NEcDsaVerifyFast`. This service uses Fast mode and Fast Modular Reduction for computation.

### 43.3.7.10.4. Parameters Definition

**Table 43-110.** GF2NEcDsaVerifyFast Service Parameters

| Parameter | Type | Direction | Location | Data Length | Before Executing the Service | After Executing the Service |
|---|---|---|---|---|---|---|
| nu1ModBase | nu1 | I | Crypto RAM | u2ModLength + 4 | Base of modulus P | Base of modulus P |
| nu1CnsBase | nu1 | I | Crypto RAM | u2ScalarLength + 8 | Base of Cns | Base of Cns |
| u2ModLength | u2 | I | – | – | Length of modulus P | Length of modulus P |
| nu1OrderPointBase | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Order of the Point A in the elliptic curve | Unchanged |
| nu1PointSignature | nu1 | I | Crypto RAM | 2*u2ScalarLength + 8 | Signature(r, s) | Corrupted |
| nu1HashBase (see **Note 1**) | nu1 | I | Crypto RAM | u2ScalarLength + 4 | Base of the hash value resulting from the previous SHA | Corrupted |
| u2ScalarLength | u2 | I | – | – | Length of scalar | Length of scalar |
| nu1PointABase | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Generator point | Corrupted |
| nu1PointPublicKeyGen | nu1 | I/O | Crypto RAM | 3*u2ModLength + 12 | Public point | Corrupted |
| nu1ABase | nu1 | I | Crypto RAM | 2*u2ModLength + 8 | Parameter a and b of the elliptic curve | Unchanged |
| nu1Workspace | nu1 | I | Crypto RAM | 8*u2ModLength + 44 | – | Corrupted workspace |

**Note:**

1. Whatever the chosen SHA, the resulting hash value may have a length inferior or equal to the modulo length and be padded with zeros until its total length is u2ModLength + 4.

### 43.3.7.10.5. Code Example

```
PUKCL_PARAM PUKCLParam;
PPUKCL_PARAM pvPUKCLParam = &PUKCLParam;

// ! The Random Number Generator must be initialized and started
// ! following the directives given for the RNG on the chip
```


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

```
PUKCL (u2Option) = 0;

// Depending on the option specified, not all fields should be filled PUKCL
_GF2NEcDsaVerify(nu1ModBase) = <Base of the ram location of P>;
PUKCL _GF2NEcDsaVerify(u2ModLength) = <Byte length of P>;
PUKCL _GF2NEcDsaVerify(nu1CnsBase) = <Base of the ram location of Cns>;
PUKCL _GF2NEcDsaVerify(nu1PointABase) = <Base of the A point>;
PUKCL _GF2NEcDsaVerify(nu1PrivateKey) = <Base of the Private Key>;
PUKCL _GF2NEcDsaVerify(nu1ScalarNumber) = <Base of the ScalarNumber>;
PUKCL _GF2NEcDsaVerify(nu1OrderPointBase) = <Base of the order of A point>;
PUKCL _GF2NEcDsaVerify(nu1ABase) = <Base of the a parameter of the curve>; PUKCL
_GF2NEcDsaVerify(nu1Workspace) = <Base of the workspace>;
PUKCL _GF2NEcDsaVerify(nu1HashBase) = <Base of the SHA resulting hash>;
...

// vPUKCL_Process() is a macro command, which populates the service name
// and then calls the library...
vPUKCL_Process(GF2NEcDsaVerifyFast, &PUKCLParam);
if (PUKCL (u2Status) == PUKCL_OK)
            {
            ...
            }
else
            if(PUKCL(u2Status) == PUKCL_WRONG_SIGNATURE)
            {
            ...
            }
else // Manage the error
```

#### 43.3.7.10.6. Constraints

No overlapping between either input and output are allowed. The following conditions must be avoided to ensure the service works correctly:

- nu1ModBase, nu1CnsBase, nu1PointABase, nu1PointPublicKeyGen, nu1PointSignature, nu1OrderPointBase,nu1ABBase, nu1Workspace or nu1HashBase are not aligned on 32-bit boundaries
- {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength + 8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PointPublicKeyGen, 3*u2ModLength + 12}, {nu1PointSignature,2*u2ScalarLength + 8}, {nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABBase, 2*u2ModLength + 8}, {nu1Workspace, <WorkspaceLength>} or {nu1HashBase, u2ScalarLength + 4} are not in Crypto RAM
- u2ModLength is either: < 12, > 0xffc or not a 32-bit length
- All overlapping between {nu1ModBase, u2ModLength + 4}, {nu1CnsBase, u2ModLength +8}, {nu1PointABase, 3*u2ModLength + 12}, {nu1PointPublicKeyGen, 3*u2ModLength + 12}, {nu1PointSignature, 2*u2ScalarLength + 8}, {nu1OrderPointBase, u2ScalarLength + 4}, {nu1ABBase, 2*u2ModLength + 8}, {nu1Workspace, <WorkspaceLength>} and {nu1HashBase, u2ScalarLength + 4}

#### 43.3.7.10.7. Status Returned Values

**Table 43-111.** GF2NEcDsaVerifyFast Service Return Codes

| Returned Status | Importance | Meaning |
|---|---|---|
| PUKCL_OK | – | The computation passed without errors. The signature is correct. |
| PUKCL_WRONG_SIGNATURE | Warning | The signature is incorrect. |

### 43.3.8. PUKCL Requirements and Performance

#### 43.3.8.1. Services Stack Usage

This library is using the main core to execute its computations, and therefore is also sharing some resources with the application.

It may be important for the application to know RAM usage by the library functions and to be aware that the library does not use any global variables.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The following table provides the minimum number of bytes used by the library that have to be available on the stacks to ensure that the functionality can be executed correctly. In some cases, the library may use less bytes than the specified number for some options. This table contains estimated values.

**Table 43-112.** Services Stack Usage

| PUKCL Service | STACK Usage (Bytes) |
|---|---|
| SelfTest | 112 |
| ClearFlags | 0 |
| Swap | 8 |
| Fill | 8 |
| CondCopy | 24 |
| FastCopy | 16 |
| Smult | 16 |
| Smult (with reduction) | 88 |
| Comp | 8 |
| Fmult | 24 |
| Fmult (with reduction) | 96 |
| Square | 16 |
| Square (with reduction) | 88 |
| Div | 144 |
| GCD | 136 |
| RedMod (Setup) | 160 |
| RedMod (using fast reduction) | 80 |
| RedMod (randomize) | 80 |
| RedMod (Normalize) | 80 |
| RedMod (Using Division) | 184 |
| ExpMod | 200 |
| PrimeGen | 416 |
| CRT | 304 |
| ZpEccAddFast | 104 |
| ZpEccAddSubFast | 92 |
| ZpEcConvProjToAffine | 280 |
| ZpEcConvAffineToProjective | 64 |
| ZpEccDblFast | 96 |
| ZpEccMulFast | 168 |
| ZpEccQuickDualMulFast | 216 |
| ZpEcDsaGenerateFast | 392 |
| ZpEcDsaVerifyFast | 456 |
| ZpEcDsaQuickVerify | 368 |
| ZpEcRandomiseCoordinate | 56 |
| GF2NEccAddFast | 128 |
| GF2NEcConvProjToAffine | 264 |
| GF2NEcConvAffineToProjective | 56 |
| GF2NEccDblFast | 136 |
| GF2NEccMulFast | 208 |
| GF2NEcDsaGenerateFast | 376 |
| GF2NEcDsaVerifyFast | 440 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-112.** Services Stack Usage (continued)

| PUKCL Service | STACK Usage (Bytes) |
|---|---|
| GF2NEcRandomiseCoordinate | 56 |

### 43.3.8.2. Parameter Size Limits for Different Services

The following table lists parameter size limits for different services.

For the services ModExp, PrimeGen, and CRT, additional details are available in the service description.

**Table 43-113.** Parameter Size Limits

| API | Min/Max Sizes | Comments |
|---|---|---|
| SelfTest | – | – |
| ClearFlags | – | – |
| Swap | 4 bytes to 2044 bytes | Per block to be swapped |
| Fill | 4 bytes to 4088 bytes | – |
| Fast Copy/Clear | 4 bytes to 2044 bytes | Supposing Length(R) = Length(X) |
| Conditional Copy/Clear | 4 bytes to 2044 bytes | Supposing Length(R) = Length(X) |
| Smult | 4 bytes to 2040 bytes | Supposing Length(R) = Length(X) + 4 Bytes, No Z Parameter, No Reduction |
| Compare | 4 bytes to 2044 bytes | Supposing Length(X) = Length(Y) |
| FMult | Input: 4 bytes to 1020 bytes Output: 4bytes to 2040 bytes | Supposing Length(Y) = Length(X), No Z Parameter, No Reduction |
| Square | Input: 4 bytes to 1020 bytes<br>Output: 4 bytes to 2040 bytes | Supposing No Z Parameter, No Reduction |
| Euclidean Division | Divider: 8 to 1016 bytes<br>Num.: 8 to 2032 bytes | Supposing Length(Num) = 2*Length(Divider) |
| Mod. inv. / GCD | 8 to 1012 bytes | – |
| ModRed | Modulus: 12 to 1016 bytes<br>Input: 24 to 2032 bytes | Supposing RBase = XBase |
| Fast ModExp Exp in Crypto RAM | 12 to 576 bytes<br>(96 to 4608 bits) | Supposing Length(Exponent) = Length(Modulus), Window Size = 1<br>With the Exponent in Crypto RAM |
| Fast ModExp<br>Exp not in Crypto RAM | 12 to 672 bytes<br>(96 to 5376 bits) | Supposing Length(Exponent) = Length(Modulus), Window Size = 1<br>With the Exponent not in Crypto RAM |
| Prime Gen. | Prime Number: 12 to 448 bytes<br>(96 to 3584 bits) | Supposing Window Size = 1 |
| CRT | Modulus = Two Primes:<br>Size of one prime from 24 to 448 bytes Modulus = from 48 to 896 bytes<br>(384 to 7168 bits) | Supposing Length(Exponent) = Length(Modulus), Window Size = 1 |
| ECC Addition qnd Subtraction GF(p) | Modulus: 12 to 308 bytes | – |
| ECC Doubling GF(p) | Modulus: 12 to 400 bytes | – |
| ECC Multiplication GF(p) | Modulus: 12 to 264 bytes | Supposing Length(Scalar) = Length(Modulus) |
| ECC Quick Dual Multiplication GF(p) | Modulus: 12 to 152 bytes | – |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-113.** Parameter Size Limits (continued)

| API | Min/Max Sizes | Comments |
|---|---|---|
| ECDSA Generate GF(p) | Modulus: 12 to 220 bytes (up to 521 bits for common curves) | Supposing Length(Scalar) = Length(Modulus) |
| ECDSA Verify GF(p) | Modulus: 12 to 188 bytes (up to 521 bits for common curves) | Supposing Length(Scalar) = Length(Modulus) |
| ECC Addition GF(2n) | Modulus: 12 to 248 bytes | – |
| ECC Doubling GF(2n) | Modulus: 12 to 364 bytes | – |
| ECC Multiplication GF(2n) | Modulus: 12 to 250bytes | Supposing Length(Scalar) = Length(Modulus) |
| ECDSA Generate GF(2n) | Modulus: 12 to 208 bytes (up to 571 bits for common curves) | Supposing Length(Scalar) = Length(Modulus) |
| ECDSA Verify GF(2n) | Modulus: 12 to 180 bytes (up to 571 bits for common curves) | Supposing Length(Scalar) = Length(Modulus) |
| ECDSA Quick Verify GF(2n) | Modulus: 12 to 140 bytes (up to 571 bits for common curves) | Supposing Length(Scalar) = Length(Modulus) |

#### 43.3.8.3. Service Timing

The values in the following tables are estimated performances for CPU clock of 120 MHz. The CPU and PUKCC are operated at the same frequency. Due to possible change in the parameters values, the measurements show approximated values.

Other test conditions:

- PUKCL library data in Crypto RAM
- Test code and test data in SRAM
- ICache and DCache are disabled

##### 43.3.8.3.1. Service Timing for RSA

RSA uses the ExpMod service for encryption and decryption. Following tables show service timing, where 'W' indicates window size.

**Table 43-114.** RSA1024

| Operation | Clock Cycles | Timing one block |
|---|---|---|
| RSA 1024 decryption / signature generation. No CRT, Regular implementation, W=4 | 3.05 MCycles | 25.42 ms |
| RSA 1024 decryption / signature generation. With CRT, Regular implementation, W=4 | 1.04 MCycles | 8.67 ms |
| RSA 1024 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=3 | 0.07 MCycles | 0.58 ms |
| RSA 1024 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=0x10001 | 0.07 MCycles | 0.58 ms |

**Table 43-115.** RSA2048

| Operation | Clock Cycles | Timing One block |
|---|---|---|
| RSA 2048 decryption / signature generation. No CRT, Regular implementation, W=4 | 21.9 MCycles | 182 ms |
| RSA 2048 decryption / signature generation. With CRT, Regular implementation, W=4 | 6.19 MCycles | 51.6 ms |
| RSA 2048 encryption / signature verification. No CRT, Fast implementation, W=1 Exponent=3 | 0.24 MCycles | 2 ms |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-115.** RSA2048 (continued)

| Operation | Clock Cycles | Timing One block |
|---|---|---|
| RSA 2048 encryption / signature verification.<br>No CRT, Fast implementation, W=1 Exponent=0x10001 | 0.24 MCycles | 2 ms |

**Table 43-116.** RSA4096

| Operation | Clock Cycles | Timing One block |
|---|---|---|
| RSA 4096 Decryption / signature generation. No CRT, Regular implementation, W=1 | 208 MCycles | 1.73s |
| RSA 4096 Decryption / signature generation. With CRT, Regular implementation, W=3 | 45.5 MCycles | 379 ms |
| RSA 4096 encryption / signature verification.<br>No CRT, Fast implementation, W=1 Exponent=3 | 0.92 MCycles | 7.67 ms |
| RSA 4096 encryption / signature verification.<br>No CRT, Fast implementation, W=1 Exponent=0x10001 | 0.92 MCycles | 7.67 ms |

#### 43.3.8.3.2. Service Timing for Prime Generation

Prime generation uses the PrimeGen service.

**Table 43-117.** Prime Generation

| Operation | Clock Cycles | Timing One Block |
|---|---|---|
| Regular Generation of two primes, Prime_Length=512 bits, W=4, Rabin Miller Iterations Number = 3, (average of 200 samples) | Mean = 47.4 MCycles | Mean = 0.40s |
| Regular Generation of two primes, Prime_Length=512 bits, W=4, Rabin Miller Iterations Number = 3, (Standard Deviation for 200 samples) | Std Dev = 30.3 Mcycles | Std Dev = 0.25s |
| Regular Generation of two primes, Prime_Length=1024 bits, W=4, Rabin Miller Iterations Number = 3, (average of 200 samples) | Mean = 448 MCycles | Mean = 3.73s |
| Regular Generation of two primes, Prime_Length=1024 bits, W=4, Rabin Miller Iterations Number = 3, (Standard Deviation for 200 samples) | Std Dev = 294 Mcycles | Std Dev = 2.45s |
| Regular Generation of two primes, Prime_Length=2048 bits, W=4, Rabin Miller Iterations Number = 3, (average of 200 samples) | Mean = 4.78 GCycles | Mean = 39.8s |
| Regular Generation of two primes, Prime_Length=2048 bits, W=4, Rabin Miller Iterations Number = 3, (Standard Deviation for 200 samples) | Std Dev = 3,05 GCycles | Std Dev = 25.4s |

#### 43.3.8.3.3. Service Timing for ECDSA on Prime Field

In the following table, ECDSA signature generation uses the ZpEcDsaGenerateFast service and signature verification uses ZpEcDsaQuickVerify

**Table 43-118.** ECDSA GF(p)

| Operation | Clock Cycles | Timing One block |
|---|---|---|
| ECDSA GF(p) 256 Generate Fast | 2.72 MCycles | 22.7 ms |
| ECDSA GF(p) 256 Verify Quick W=(6,6)<br>Scalar in Classical RAM | 1.78 MCycles | 14.8 ms |
| ECDSA GF(p) 256 Verify Quick W=(4,4)<br>Scalar in PUKCC RAM | 1.83 MCycles | 15.2 ms |
| ECDSA GF(p) 384 Generate Fast | 6.28 MCycles | 52.3 ms |
| ECDSA GF(p) 384 Verify Quick W=(5,5)<br>Scalar in Classical RAM | 3.93 MCycles | 32.8 ms |
| ECDSA GF(p) 384 Verify Quick W=(4,4)<br>Scalar in PUKCC RAM | 4.09 MCycles | 34.1 ms |
| ECDSA GF(p) 521 Generate Fast | 13.4 MCycles | 112 ms |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 43-118.** ECDSA GF(p) (continued)

| Operation | Clock Cycles | Timing One block |
|---|---|---|
| ECDSA GF(p) 521 Verify Quick W=(4,5)<br>Scalar in Classical RAM | 8.4 MCycles | 70.3 ms |
| ECDSA GF(p) 521 Verify Quick W=(4,4)<br>Scalar in PUKCC RAM | 8.6 MCycles | 72ms |

#### 43.3.8.3.4. Service Timing for ECDSA on Binary Field

In the following table, ECDSA signature generation uses the GF2NEcDsaGenerateFast service and signature verification uses GF2NEcDsaVerifyFast

**Table 43-119.** ECDSA GF($2^n$)

| Operation | CPU Cycles | Timing One block |
|---|---|---|
| ECDSA GF($2^n$) B283 Generate Fast | 3.21 MCycles | 26.8 ms |
| ECDSA GF($2^n$) B283 Verify | 6.44 MCycles | 53.5 ms |
| ECDSA GF($2^n$) B409 Generate Fast | 6.93 Mcycles | 57.8 ms |
| ECDSA GF($2^n$) B409 Verify | 13.8 Mcycles | 115 ms |
| ECDSA GF($2^n$) B571 Generate Fast | 15.1 Mcycles | 125 ms |
| ECDSA GF($2^n$) B571 Verify | 30.1 MCycles | 251 ms |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 44. TRNG – True Random Number Generator

## 44.1. Overview

The True Random Number Generator (TRNG) generates unpredictable random numbers that are not generated by an algorithm. It passes the American NIST Special Publication 800-22 and Diehard Random Tests Suites.

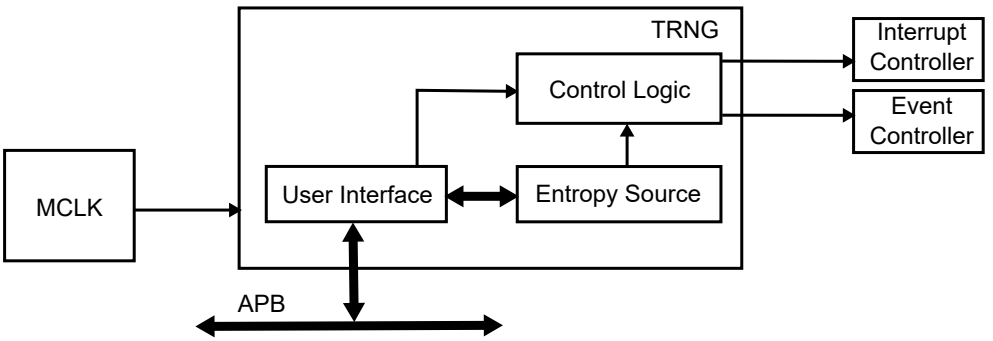
The TRNG may be used as an entropy source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3.

## 44.2. Features

- Passed NIST Special Publication 800-22 Tests Suite
- Passed Diehard Random Tests Suite
- May be used as Entropy Source for seeding an NIST approved DRNG (Deterministic RNG) as required by FIPS PUB 140-2 and 140-3
- Provides a 32-bit random number every 84 clock cycles

## 44.3. Block Diagram

**Figure 44-1.** TRNG Block Diagram.




## 44.4. Signal Description

Not applicable.

## 44.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 44.5.1. I/O Lines

Not applicable.

### 44.5.2. Power Management

The functioning of TRNG depends on the sleep mode of device.

The TRNG interrupts can be used to wake up the device from sleep modes. Events connected to the event system can trigger other operations in the system without exiting sleep modes.

**Related Links**

PM – Power Manager

Sleep Mode Operation


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 44.5.3. Clocks

The TRNG bus clock (CLK_TRNG_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_TRNG_APB can be found in *Peripheral Clock Masking*.

**Related Links**
Peripheral Clock Masking

### 44.5.4. DMA

Not applicable.

### 44.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the TRNG interrupt(s) requires the interrupt controller to be configured first. Refer to NVIC - Nested Interrupt *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller

### 44.5.6. Events

TRNG can generate Events that are used by the Event System (EVSYS) and EVSYS users.

TRNG cannot use any Events from other peripherals, as it is not an Event User.

**Related Links**
Event System (EVSYS)

### 44.5.7. Debug Operation

When the CPU is halted in debug mode the TRNG continues normal operation. If the TRNG is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

### 44.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the following register:

Interrupt Flag Status and Clear (INTFLAG) register

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

### 44.5.9. Analog Connections

Not applicable.

## 44.6. Functional Description

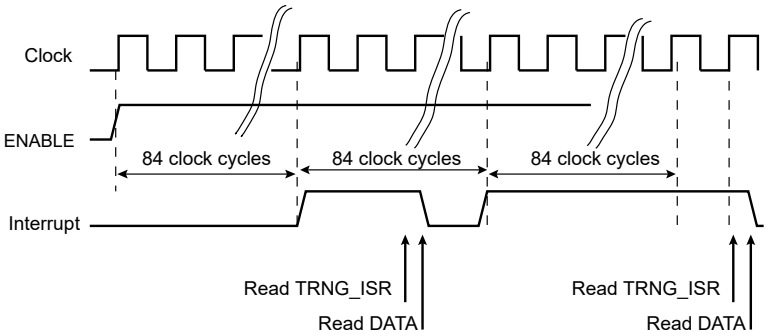
### 44.6.1. Principle of Operation

When the TRNG is enabled, the peripheral starts providing new 32-bit random numbers every 84 CLK_TRNG_APB clock cycles.

The TRNG can be configured to generate an interrupt or event when a new random number is available.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 44-2.** TRNG Data Generation Sequence




## 44.6.2. Basic Operation

### 44.6.2.1. Initialization

To operate the TRNG, do the following:

- Configure the clock source for CLK_TRNG_APB in the Main Clock Controller (MCLK) and enable the clock by writing a '1' to the TRNG bit in the APB Mask register of the MCLK.
- Optional: Enable the output event by writing a '1' to the EVCTRL.DATARDYEO bit.
- Optional: Enable the TRNG to Run in Standby sleep mode by writing a '1' to CTRLA.RUNSTDBY.
- Enable the TRNG operation by writing a '1' to CTRLA.ENABLE.

The following register is enable-protected, meaning that it can only be written when the TRNG is disabled (CTRLA.ENABLE is zero):

- Event Control register (EVCTRL)

Enable-protection is denoted by the Enable-Protected property in the register description.

### 44.6.2.2. Enabling, Disabling and Resetting

The TRNG is enabled by writing '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TRNG is disabled by writing a zero to CTRLA.ENABLE.

## 44.6.3. Interrupts

The TRNG has the following interrupt source:

- Data Ready (DATARDY): Indicates that a new random number is available in the DATA register and ready to be read.
  This interrupt is a synchronous wake-up source. See *Sleep Mode Controller* for details.

The interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.DATARDY) is set to '1' when the interrupt condition occurs. The interrupt can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET.DATARDY), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, or the interrupt is disabled. See INTFLAG for details on how to clear interrupt flags.

Note that interrupts must be globally enabled for interrupt requests to be generated.

**Related Links**

Nested Vector Interrupt Controller

## 44.6.4. Events

The TRNG can generate the following output event:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Data Ready (DATARDY): Generated when a new random number is available in the DATA register.

Writing '1' to the Data Ready Event Output bit in the Event Control Register (EVCTRL.DATARDYEO) enables the DTARDY event. Writing a '0' to this bit disables the corresponding output event. Refer to *EVSYS – Event System* for details on configuring the Event System.

**Related Links**

Event System (EVSYS)

### 44.6.5. Sleep Mode Operation

The Run in Standby bit in Control A register (CTRLA.RUNSTDBY) controls the behavior of the TRNG during standby sleep mode:

When this bit is '0', the TRNG is disabled during sleep, but maintains its current configuration.

When this bit is '1', the TRNG continues to operate during sleep and any enabled TRNG interrupt source can wake up the CPU.

### 44.6.6. Synchronization

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 44.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | RUNSTDBY | | | | | ENABLE | |
| 0x01 ... 0x03 | Reserved | | | | | | | | | |
| 0x04 | EVCTRL | 7:0 | | | | | | | | DATARDYEO |
| 0x05 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | | | | | | | | DATARDY |
| 0x09 | INTENSET | 7:0 | | | | | | | | DATARDY |
| 0x0A | INTFLAG | 7:0 | | | | | | | | DATARDY |
| 0x0B ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | DATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |

## 44.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.

Refer to *PAC - Peripheral Access Controller* and Synchronizationfor details.

**Related Links**
PAC - Peripheral Access Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 44.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | RUNSTDBY | | | | | ENABLE | |
| Access | | R/W | | | | | R/W | |
| Reset | | 0 | | | | | 0 | |

**Bit 6 – RUNSTDBY** Run in Standby
This bit controls how the ADC behaves during standby sleep mode:

| Value | Description |
|---|---|
| 0 | The TRNG is halted during standby sleep mode. |
| 1 | The TRNG is not stopped in standby sleep mode. |

**Bit 1 – ENABLE** Enable

| Value | Description |
|---|---|
| 0 | The TRNG is disabled. |
| 1 | The TRNG is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 44.8.2. Event Control

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DATARDYEO |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DATARDYEO** Data Ready Event Output
This bit indicates whether the Data Ready event output is enabled and whether an output event will be generated when a new random value is ready.

| Value | Description |
|---|---|
| 0 | Data Ready event output is disabled and an event will not be generated. |
| 1 | Data Ready event output is enabled and an event will be generated. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 44.8.3. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DATARDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DATARDY** Data Ready Interrupt Enable
Writing a '1' to this bit will clear the Data Ready Interrupt Enable bit, which disables the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The DATARDY interrupt is disabled. |
| 1 | The DATARDY interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 44.8.4. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DATARDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DATARDY** Data Ready Interrupt Enable
Writing a '1' to this bit will set the Data Ready Interrupt Enable bit, which enables the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The DATARDY interrupt is disabled. |
| 1 | The DATARDY interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 44.8.5. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DATARDY |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DATARDY** Data Ready
This flag is set when a new random value is generated, and an interrupt will be generated if INTENCLR/SET.DATARDY=1.
This flag is cleared by writing a '1' to the flag or by reading the DATA register.
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 44.8.6. Output Data

**Name:** DATA
**Offset:** 0x20
**Reset:** 0xXXXXXXXX
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | x | x | x | x | x | x | x | x |

**Bits 31:0 – DATA[31:0]** Output Data
These bits hold the 32-bit randomly generated output data.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 45. ADC – Analog-to-Digital Converter

## 45.1. Overview

The Analog-to-Digital Converter (ADC) converts analog signals to digital values. The ADC has up to 12-bit resolution, and is capable of a sampling rate of up to 1MSPS. The input selection is flexible, and both differential and single-ended measurements can be performed. In addition, several internal signal inputs are available. The ADC can provide both signed and unsigned results.

ADC measurements can be started by either application software or an incoming event from another peripheral in the device. ADC measurements can be started with predictable timing, and without software intervention.

Both internal and external reference voltages can be used.

An integrated temperature sensor is available for use with the ADC. The bandgap voltage, as well as the scaled I/O and core voltages, can also be measured by the ADC.

The ADC has a compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required.

The ADC can be configured for 8-, 10- or 12-bit results. ADC conversion results are provided left- or right-adjusted, which eases calculation when the result is represented as a signed value. It is possible to use DMA to move ADC results directly to memory or peripherals when conversions are done.

The SAM D5x/E5x has two ADC instances, ADC0 and ADC1. The two inputs can be sampled simultaneously, as each ADC includes sample and hold circuits.

**Note:** When the Peripheral Touch Controller (PTC) is enabled, ADC0 is serving the PTC exclusively. In this case, ADC0 cannot be used by the user application.

## 45.2. Features

The following are key features of the ADC:

- Two Analog-to-Digital Converters (ADC): ADC0 and ADC1
- 8-bit, 10-bit, or 12-bit resolution
- Up to 1,000,000 samples per second (1 Msps)
- Differential -ended and single-ended inputs
  - Up to 23 analog inputs per ADC (16 external channels)
    23 positive and 8 negative, including internal and external
- Internal inputs:
  - Internal temperature sensor
  - Scaled core supply
  - Scaled I/O supply
  - Scaled VBAT supply
  - DAC0
- Single, continuous, and sequencing options
- Windowing monitor with selectable channel
- Conversion range: $V_{ref}$ = [1.0V to VDDANA]
- Built-in internal reference and external reference options
- Event-triggered conversion for accurate timing (one event input)

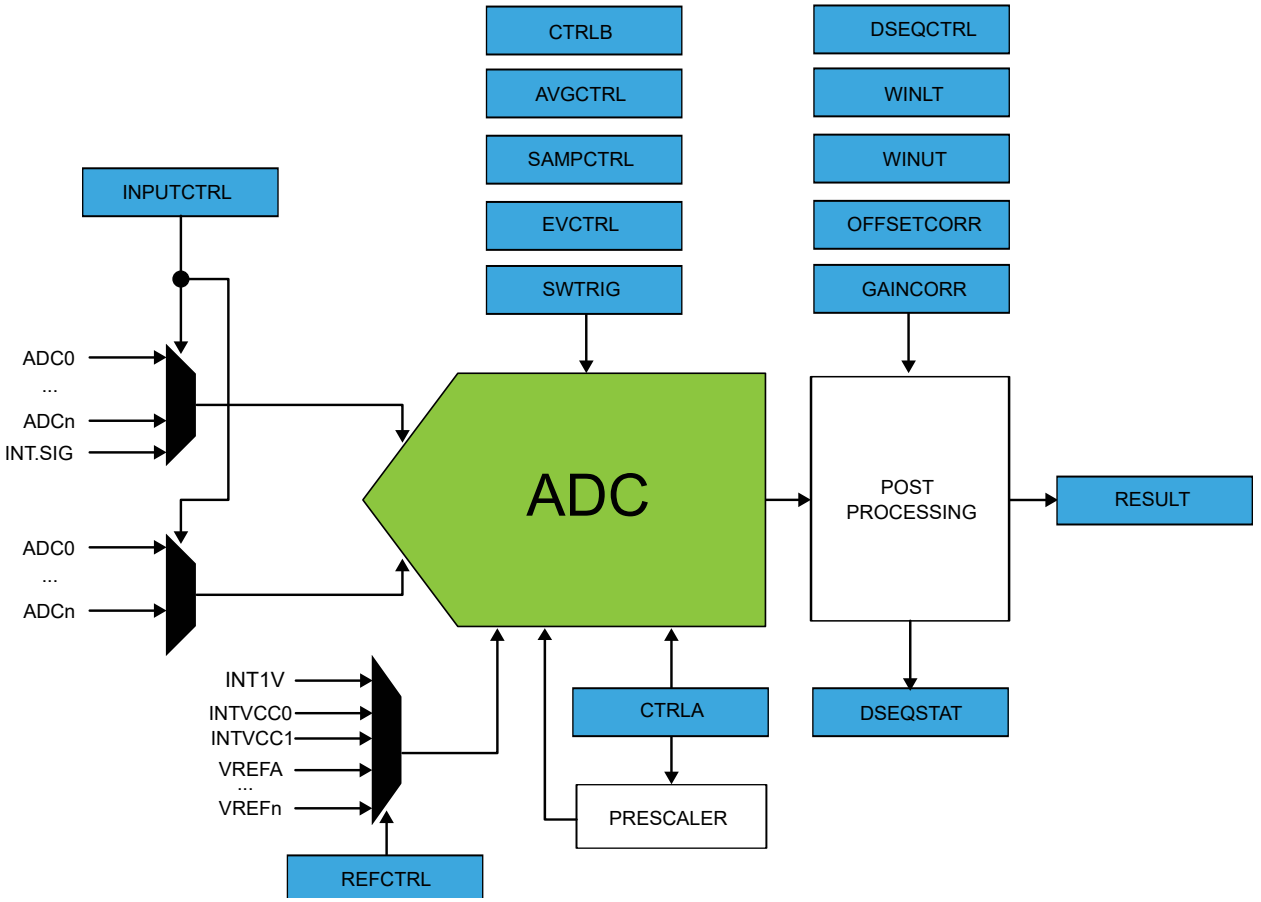

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Optional DMA transfer of conversion settings or result
- Hardware gain and offset compensation
- Averaging and oversampling with decimation to support up to 16-bit result
- Selectable sampling time
- Flexible Power or Throughput Rate Management

ADC0 can be configured to serve the Peripheral Touch Controller (PTC). Refer to the *PTC* chapter for additional information.

## 45.3. Block Diagram

**Figure 45-1.** ADC Block Diagram




## 45.4. Signal Description

| Signal | Description | Type |
|---|---|---|
| VREF[A, B, C] | Analog input | External reference voltage |
| AIN[15..0] | Analog input | Analog input channels |

**Note:** One signal can be mapped on several pins.

**Related Links**

I/O Multiplexing and Considerations


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 45.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 45.5.1. I/O Lines

Using the ADC's I/O lines requires the I/O pins to be configured using the port configuration (PORT).

**Related Links**

PORT - I/O Pin Controller

### 45.5.2. Power Management

The ADC will continue to operate in any Sleep mode where the selected source clock is running. The ADC's interrupts, except the OVERRUN interrupt, can be used to wake up the device from Sleep modes, except the OVERRUN interrupt. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

**Related Links**

PM – Power Manager

### 45.5.3. Clocks

The ADC bus clocks (CLK_APB_ADCx) can be enabled in the Main Clock, which also defines the default state.

Each ADC requires a generic clock (GCLK_ADCx). This clock must be configured and enabled in the Generic Clock Controller (GCLK) before using the ADC.

A generic clock is asynchronous to the bus clock. Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to *Synchronization* for further details.

**Related Links**

Peripheral Clock Masking
GCLK - Generic Clock Controller

### 45.5.4. DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the ADC DMA requests requires the DMA Controller to be configured first.

**Related Links**

DMAC – Direct Memory Access Controller

### 45.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the ADC interrupt requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 45.5.6. Events

The events are connected to the Event System.

**Related Links**

Event System (EVSYS)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.5.7. Debug Operation

When the CPU is halted in Debug mode the ADC will halt normal operation. The ADC can be forced to continue operation during debugging. Refer to the DBGCTRL register for additional information.

**Related Links**

DBGCTRL

Debug Control

### 45.5.8. Register Access Protection

All registers with write-access are optionally write-protected by the Peripheral Access Controller (PAC), except the Interrupt Flag Status and Clear (INTFLAG) register.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**

PAC - Peripheral Access Controller

### 45.5.9. Analog Connections

The I/O pins (AINx) and the VREFA/VREFB/VREFC reference voltage pins are analog inputs to the ADC. Any internal reference source, such as a bandgap voltage reference or DAC must be configured and enabled prior to its use with the ADC.

### 45.5.10. Calibration

The BIASREFBUF, BIASR2R and BIASCOMP calibration values from the production test must be loaded from the NVM Software Calibration Area into the ADC Calibration register (CALIB) by software to allow conversion and to achieve specified accuracy.

## 45.6. Functional Description

### 45.6.1. Principle of Operation

By default, the ADC provides results with 12-bit resolution. 8-bit or 10-bit results can be selected in order to reduce the conversion time, see Conversion Timing and Sampling Rate.

The ADC has an oversampling with decimation option that can extend the resolution to 16 bits. The input values can be either internal (e.g., an internal temperature sensor) or external (connected I/O pins). The user can also configure whether the conversion should be single-ended or differential.

### 45.6.2. Basic Operation

#### 45.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the ADC is disabled (CTRLA.ENABLE = 0):

- The Control A (CTRLA) register, except the ENABLE and SWRST bits
- The Event Control register (EVCTRL)
- The Calibration register (CALIB)

Enable-protection is denoted by the ‘Enable-Protected’ property in the register description.

#### 45.6.2.2. Enabling, Disabling, and Resetting

The ADC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The ADC is disabled by writing CTRLA.ENABLE=0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The ADC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the ADC, except DBGCTRL, will be reset to their initial state, and the ADC will be disabled. Refer to CTRLA for details.

### 45.6.2.3. Operation

In the most basic configuration, the ADC samples values from the configured internal or external sources (INPUTCTRL register). The rate of the conversion depends on the combination of the GCLK_ADCx frequency and the clock prescaler.

To convert analog values to digital values, the ADC needs to be initialized first, as described in the Initialization section. Data conversion can be started either manually by setting the Start bit in the Software Trigger register (SWTRIG.START=1), or automatically by configuring an automatic trigger to initiate the conversions. The ADC starts sampling the input only after the start of conversion is triggered. This means that even after the MUX selection is made, sample and hold (S&H) operation starts only on the conversion trigger. A free-running mode can be used to continuously convert an input channel. When using free-running mode the first conversion must be started, while subsequent conversions will start automatically at the end of previous conversions.

The ADC starts sampling the input only after the start of a conversion is triggered. This means that even after the MUX selection is made, sample and hold operation starts only on the conversion trigger.

The result of the conversion is stored in the Result register (RESULT) overwriting the result from the previous conversion.

To avoid data loss, if more than one channel is enabled, the conversion result must be read as soon as it is available (INTFLAG.RESRDY). Failing to do so will result in an overrun error condition, indicated by the OVERRUN bit in the Interrupt Flag Status and Clear register (INTFLAG.OVERRUN).

To enable one of the available interrupts sources, the corresponding bit in the Interrupt Enable Set register (INTENSET) must be written to '1'.

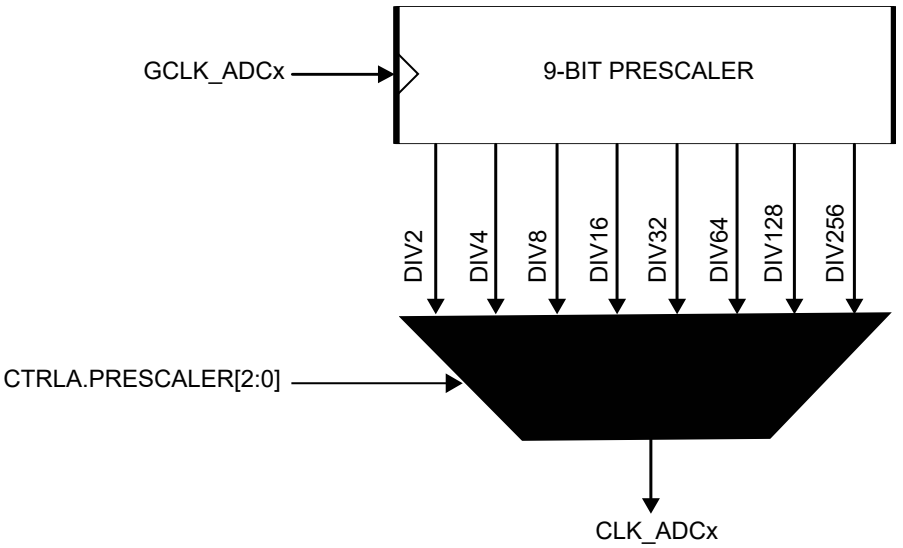
**Related Links**

Initialization

### 45.6.2.4. Prescaler Selection

The ADC is clocked by GCLK_ADCx. There is also a prescaler in the ADC to enable conversion at lower clock rates. Refer to CTRLA for details on prescaler settings. Refer to Conversion Timing and Sampling Rate for details on timing and sampling rate.

**Figure 45-2.** ADC Prescaler





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** The minimum prescaling factor is DIV2.

### 45.6.2.5. Reference Configuration

The ADC has various sources for its reference voltage $V_{REF}$. The Reference Voltage Selection bit field in the Reference Control register (REFCTRL.REFSEL) determines which reference is selected. By default, the internal voltage reference VREF is selected. Based on customer application requirements, the external or internal reference can be selected. Refer to REFCTRL.REFSEL for further details on available selections.

**Related Links**

REFCTRL

Reference Control

### 45.6.2.6. ADC Resolution

The ADC supports 8-bit, 10-bit or 12-bit resolution. Resolution can be changed by writing the Resolution bit group in the Control B register (CTRLB.RESSEL). By default, the ADC resolution is set to 12 bits. The resolution affects the propagation delay, see also Conversion Timing and Sampling Rate.

### 45.6.2.7. Differential-ended and Single-Ended Conversions

The ADC has two conversion options: differential-ended and single-ended.

If the positive input is always positive, the single-ended conversion should be used in order to have full 12-bit resolution in the conversion.

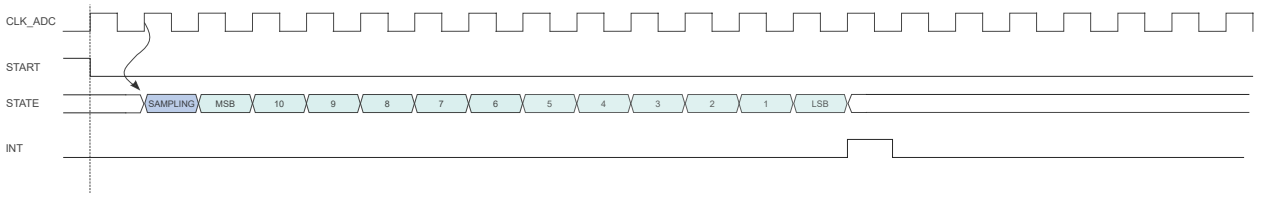
If the positive input level may go below the negative input, the differential mode should be used in order to get correct results.

The differential mode is enabled by writing a '1' to the DIFFMODE bit in the input control register (INPUTCTRL.DIFFMODE). Both conversion types could be run in single mode or in free-running mode. When the free-running mode is selected, an ADC input will continuously sample the input and performs a new conversion. The INTFLAG.RESRDY bit will be set at the end of each conversion.

### 45.6.2.8. Conversion Timing and Sampling Rate

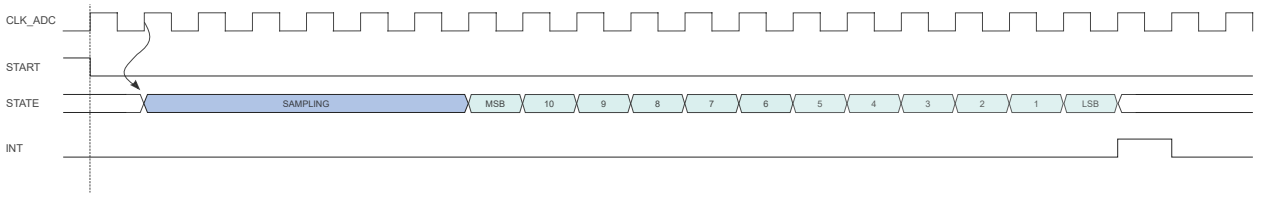
The following figure shows the ADC timing for one single conversion. A conversion starts after the software or event start are synchronized with the GCLK_ADCx clock. The input channel is sampled in the first half CLK_ADCx period.

**Figure 45-3.** ADC Timing for One Conversion in 12-bit Resolution




The sampling time can be increased by using the Sampling Time Length bit group in the Sampling Time Control register (SAMPCTRL.SAMPLEN). The figure below illustrate the timing conversion with sampling time increased to six CLK_ADC cycles.

**Figure 45-4.** ADC Timing for One Conversion with Increased Sampling Time, 12-bit





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Also, the ADC provides offset compensation, see the following figure. The offset compensation is enabled by the Offset Compensation bit in the Sampling Control register (SAMPCTRL.OFFCOMP).

**Note:** If offset compensation is used, the sampling time must be set to one cycle of CLK_ADCx.

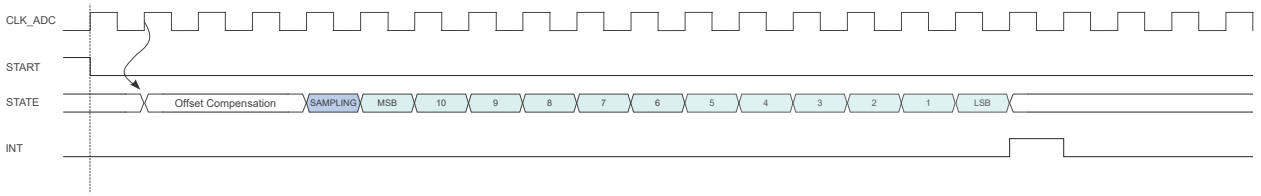
In free running mode, the sampling rate $R_S$ is calculated by:

$R_S = f_{CLK_ADC} / ( n_{SAMPLING} + n_{OFFCOMP} + n_{DATA})$

Where,

- $n_{SAMPLING}$ is the sampling duration in CLK_ADC cycles
- $n_{OFFCOMP}$ is the offset compensation duration in clock cycles
- $n_{DATA}$ is the bit resolution
- $f_{CLK_ADC}$ is the ADC clock frequency from the internal prescaler: $f_{CLK_ADC} = f_{GCLK_ADC} / 2^{(1 + CTRLA.PRESCALER)}$

**Figure 45-5.** ADC Timing for One Conversion with Offset Compensation, 12-bit




The impact of resolution on the sampling rate is seen in the following two figures, where free-running sampling in 12-bit and 8-bit resolution are compared.

**Figure 45-6.** ADC Timing for Free Running in 12-bit Resolution

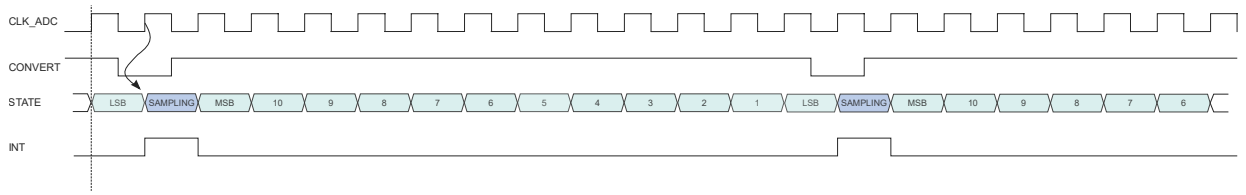
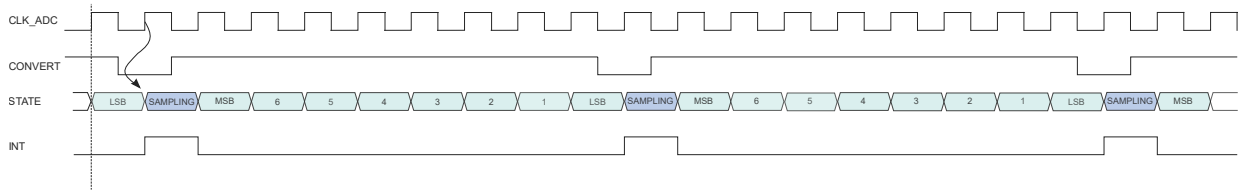



**Figure 45-7.** ADC Timing for Free Running in 8-bit Resolution




The propagation delay of an ADC measurement is given by:

$$\text{PropagationDelay} = \frac{1 + \text{Resolution}}{f_{\text{ADC}}}$$

**Example.** To obtain 1 Msps in 12-bit resolution with a sampling time length of four CLK_ADC cycles, $f_{CLK_ADC}$ must be 1 Msps * (4 + 12) = 16 MHz. As the minimal division factor of the prescaler is 2, GCLK_ADC must be 32 MHz.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.6.2.9. Accumulation

The results of multiple, consecutive conversions can be accumulated. The number of samples to be accumulated is specified by the Sample Number field in the Average Control register (AVGCTRL.SAMPLENUM). When accumulating more than 16 samples, the result will be too large to fit the 16-bit RESULT register size. To avoid overflow, the result is right shifted automatically to fit within the available register size. The number of automatic right shifts is specified in the table below.

**Note:** To perform the accumulation of two or more samples, the Conversion Result Resolution field in the Control B register (CTRLB.RESSEL) must be set.

**Table 45-1.** Accumulation

| Number of Accumulated Samples | AVGCTRL. SAMPLENUM | Number of Automatic Right Shifts | Final Result Precision | Automatic Division Factor |
|---|---|---|---|---|
| 1 | 0x0 | 0 | 12 bits | 0 |
| 2 | 0x1 | 0 | 13 bits | 0 |
| 4 | 0x2 | 0 | 14 bits | 0 |
| 8 | 0x3 | 0 | 15 bits | 0 |
| 16 | 0x4 | 0 | 16 bits | 0 |
| 32 | 0x5 | 1 | 16 bits | 2 |
| 64 | 0x6 | 2 | 16 bits | 4 |
| 128 | 0x7 | 3 | 16 bits | 8 |
| 256 | 0x8 | 4 | 16 bits | 16 |
| 512 | 0x9 | 5 | 16 bits | 32 |
| 1024 | 0xA | 6 | 16 bits | 64 |
| Reserved | 0xB –0xF | | 12 bits | 0 |

**Table 45-2.** Accumulation

| Number of Accumulated Samples | AVGCTRL. SAMPLENUM | Intermediate Result Precision | Number of Automatic Right Shifts | Final Result Precision | Automatic Division Factor |
|---|---|---|---|---|---|
| 1 | 0x0 | 12 bits | 0 | 12 bits | 0 |
| 2 | 0x1 | 13 bits | 0 | 13 bits | 0 |
| 4 | 0x2 | 14 bits | 0 | 14 bits | 0 |
| 8 | 0x3 | 15 bits | 0 | 15 bits | 0 |
| 16 | 0x4 | 16 bits | 0 | 16 bits | 0 |
| 32 | 0x5 | 17 bits | 1 | 16 bits | 2 |
| 64 | 0x6 | 18 bits | 2 | 16 bits | 4 |
| 128 | 0x7 | 19 bits | 3 | 16 bits | 8 |
| 256 | 0x8 | 20 bits | 4 | 16 bits | 16 |
| 512 | 0x9 | 21 bits | 5 | 16 bits | 32 |
| 1024 | 0xA | 22 bits | 6 | 16 bits | 64 |
| Reserved | 0xB –0xF | 12 bits | | 12 bits | 0 |

### 45.6.2.10. Averaging

Averaging is a feature that increases the sample accuracy, at the cost of a reduced sampling rate. This feature is suitable when operating in noisy conditions.

Averaging is done by accumulating m samples, as described in Accumulation, and dividing the result by m. The averaged result is available in the RESULT register. The number of samples to be accumulated is specified by writing to AVGCTRL.SAMPLENUM as shown in Table 45-3.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The division is obtained by a combination of the automatic right shift described above, and an additional right shift that must be specified by writing to the Adjusting Result/Division Coefficient field in AVGCTRL (AVGCTRL.ADJRES), as described in Table 45-3.

**Note:** To perform the averaging of two or more samples, the Conversion Result Resolution field in the Control B register (CTRLB.RESSEL) must be set.

Averaging AVGCTRL.SAMPLENUM samples will reduce the un-averaged sampling rate by a factor $\frac{1}{\text{AVGCTRL.SAMPLENUM}}$.

When the averaged result is available, the INTFLAG.RESRDY bit will be set.

**Table 45-3.** Averaging

| Number of Accumulated Samples | AVGCTRL.SAMPLENUM | Intermediate Result Precision | Number of Automatic Right Shifts | Division Factor | AVGCTRL.ADJRES | Total Number of Right Shifts | Final Result Precision | Automatic Division Factor |
|---|---|---|---|---|---|---|---|---|
| 1 | 0x0 | 12 bits | 0 | 1 | 0x0 | | 12 bits | 0 |
| 2 | 0x1 | 13 | 0 | 2 | 0x1 | 1 | 12 bits | 0 |
| 4 | 0x2 | 14 | 0 | 4 | 0x2 | 2 | 12 bits | 0 |
| 8 | 0x3 | 15 | 0 | 8 | 0x3 | 3 | 12 bits | 0 |
| 16 | 0x4 | 16 | 0 | 16 | 0x4 | 4 | 12 bits | 0 |
| 32 | 0x5 | 17 | 1 | 16 | 0x4 | 5 | 12 bits | 2 |
| 64 | 0x6 | 18 | 2 | 16 | 0x4 | 6 | 12 bits | 4 |
| 128 | 0x7 | 19 | 3 | 16 | 0x4 | 7 | 12 bits | 8 |
| 256 | 0x8 | 20 | 4 | 16 | 0x4 | 8 | 12 bits | 16 |
| 512 | 0x9 | 21 | 5 | 16 | 0x4 | 9 | 12 bits | 32 |
| 1024 | 0xA | 22 | 6 | 16 | 0x4 | 10 | 12 bits | 64 |
| Reserved | 0xB –0xF | | | | 0x0 | | 12 bits | 0 |

### 45.6.2.11. Oversampling and Decimation

By using oversampling and decimation, the ADC resolution can be increased from 12 bits up to 16 bits, for the cost of reduced effective sampling rate.

To increase the resolution by n bits, $4^n$ samples must be accumulated. The result must then be right shifted by n bits. This right shift is a combination of the automatic right shift and the value written to AVGCTRL.ADJRES. To obtain the correct resolution, the ADJRES must be configured as described in the table below. This method will result in n bit extra LSB resolution.

**Table 45-4.** Configuration Required for Oversampling and Decimation

| Result Resolution | Number of Samples to Average | AVGCTRL.SAMPLENUM[3:0] | Number of Automatic Right Shifts | AVGCTRL.ADJRES[2:0] |
|---|---|---|---|---|
| 13 bits | $4^1 = 4$ | 0x2 | 0 | 0x1 |
| 14 bits | $4^2 = 16$ | 0x4 | 0 | 0x2 |
| 15 bits | $4^3 = 64$ | 0x6 | 2 | 0x1 |
| 16 bits | $4^4 = 256$ | 0x8 | 4 | 0x0 |

### 45.6.2.12. Window Monitor

The window monitor feature allows comparing the conversion result in the RESULT register to predefined threshold values.

The window mode is selected by writing the Window Monitor Mode bits in the Control B register (CTRLB.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT) and Window Monitor Upper Threshold register (WINUT).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When the Window Single Sample (CTRLB.WINSS) bit is written to '1', the window comparator is working on each sample instead of the accumulated value. The number of samples matching with window comparator is available on Window Comparator Counter bits (STATUS.WCC).

In differential mode, WINLT and WINUT are evaluated as signed values. Otherwise they are evaluated as unsigned values. The significant WINLT and WINUT bits are given by the precision selected in the Conversion Result Resolution bit group in the Control B register (CTRLB.RESSEL). This means that for example in 8-bit mode, only the eight lower bits will be considered. In addition, in differential mode, the eighth bit will be considered as the sign bit, even if the ninth bit is zero.

The INTFLAG.WINMON interrupt flag is set when either the conversion result matches the window monitor condition, when the Window Comparator Counter is not zero in case of accumulation with CTRLB.WINSS=1.

#### 45.6.2.13. Offset and Gain Correction

Inherent gain and offset errors affect the absolute accuracy of the ADC.

The offset error is defined as the deviation of the actual ADC transfer function from an ideal straight line at zero input voltage. The offset error cancellation is handled by the Offset Correction register (OFFSETCORR). The offset correction value is subtracted from the converted data before writing the Result register (RESULT).

The gain error is defined as the deviation of the last output step's midpoint from the ideal straight line, after compensating for offset error. The gain error cancellation is handled by the Gain Correction register (GAINCORR).

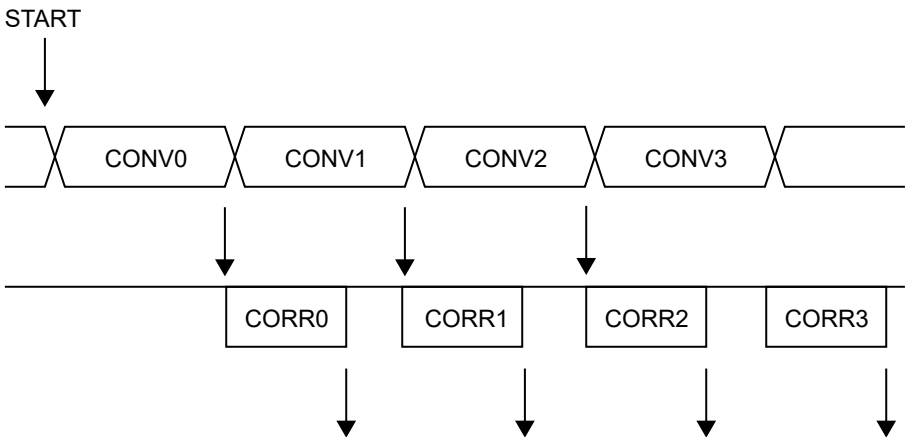
To correct these two errors, the Digital Correction Logic Enabled bit in the Control B register (CTRLB.CORREN) must be set.

Offset and gain error compensation results are both calculated according to:

$$\text{Result} = (\text{Conversion value} + - \text{OFFSETCORR}) \cdot \text{GAINCORR}$$

The correction will introduce a latency of 13 CLK_ADC clock cycles. In free running mode this latency is introduced on the first conversion only, since its duration is always less than the propagation delay. In single conversion mode this latency is introduced for each conversion.

**Figure 45-8.** ADC Timing Correction Enabled




### 45.6.3. Additional Features

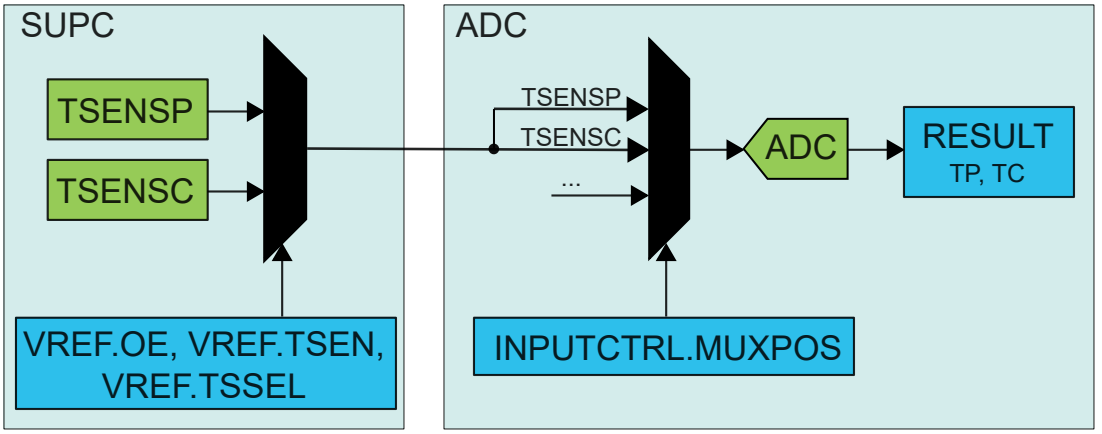
#### 45.6.3.1. Device Temperature Measurement

The device provides two temperature sensors (TSENSP and TSENSC, respectively) at different locations in the die, controlled by the SUPC - Supply Controller. The output voltages from the sensors, $V_{TP}$ and $V_{TC}$, can be sampled by the ADC.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The respective temperature sensor selection is dependent on the configuration of SUPC:

- If the SUPC is not in on-demand mode (SUPC.VREF.ONDEMAND=0), and if SUPC.VREF.TSEN=1 and SUPC.VREF.VREFOE=0, the temperature sensor is selected by writing to the Temperature Sensor Channel Selection bit in the Voltage Reference System Control register (SUPC.VREF.TSSEL).




  The state of the MUX input selection bit fields in the ADC Input Control register (ADC.INPUTCTRL.MUXPOS and MUXNEG) does not affect the sensor selection.

- If the SUPC is in on-demand mode in (SUPC.VREF.ONDEMAND=1) and SUPC.VREF.TSEN=1, the output will be automatically set to the sensor requested by the ADC, independent of SUPC.VREF.TSSEL. SUPC.VREF.VREFOE can also be set to '1'.

  Which temperature sensor is requested by the ADC is selected by writing to the Positive MUX Input Selection bits in the Input Control register (ADC.INPUTCTRL.MUXPOS).

Using the two conversion results, TP and TC, and the temperature calibration parameters found in the NVM Software Calibration Area, the die temperature *T* can be calculated:

$$T = \frac{\mathrm{TL} \cdot \mathrm{VPH} \cdot \mathrm{TC} - \mathrm{VPL} \cdot \mathrm{TH} \cdot \mathrm{TC} - \mathrm{TL} \cdot \mathrm{VCH} \cdot \mathrm{TP} + \mathrm{TH} \cdot \mathrm{VCL} \cdot \mathrm{TP}}{\mathrm{VCL} \cdot \mathrm{TP} - \mathrm{VCH} \cdot \mathrm{TP} - \mathrm{VPL} \cdot \mathrm{TC} + \mathrm{VPH} \cdot \mathrm{TC}}$$

Here, TL and TH are decimal numbers composed of their respective integer part (TLI, THI) and decimal parts (TLD and THD) from the NVM Software Calibration Area.

**Note:** The accuracy is dependent on the current temperature, and degrades towards extreme conditions.

**Related Links**

SUPC – Supply Controller

### 45.6.3.2. Rail-to-Rail Operation

The accuracy of the ADC is highest when the input common mode voltage (VCMIN) is close to VREF/2. To enable a full range of common mode voltages (rail-to-rail operation), the Rail-to-Rail bit in the Control C register (CTRLA.R2R) must be written to one. The rail-to-rail operation requires a sampling period of four cycles. This is achieved by enabling offset compensation (SAMPCTRL.OFFCOMP = 1). The rail-to-rail operation should not be used when offset compensation is disabled.

### 45.6.3.3. Double Buffering

The following registers are double buffered:

- Input Control (INPUTCTRL)
- Control B (CTRLB)
- Reference Control (REFCTRL)
- Average Control (AVGCTRL)

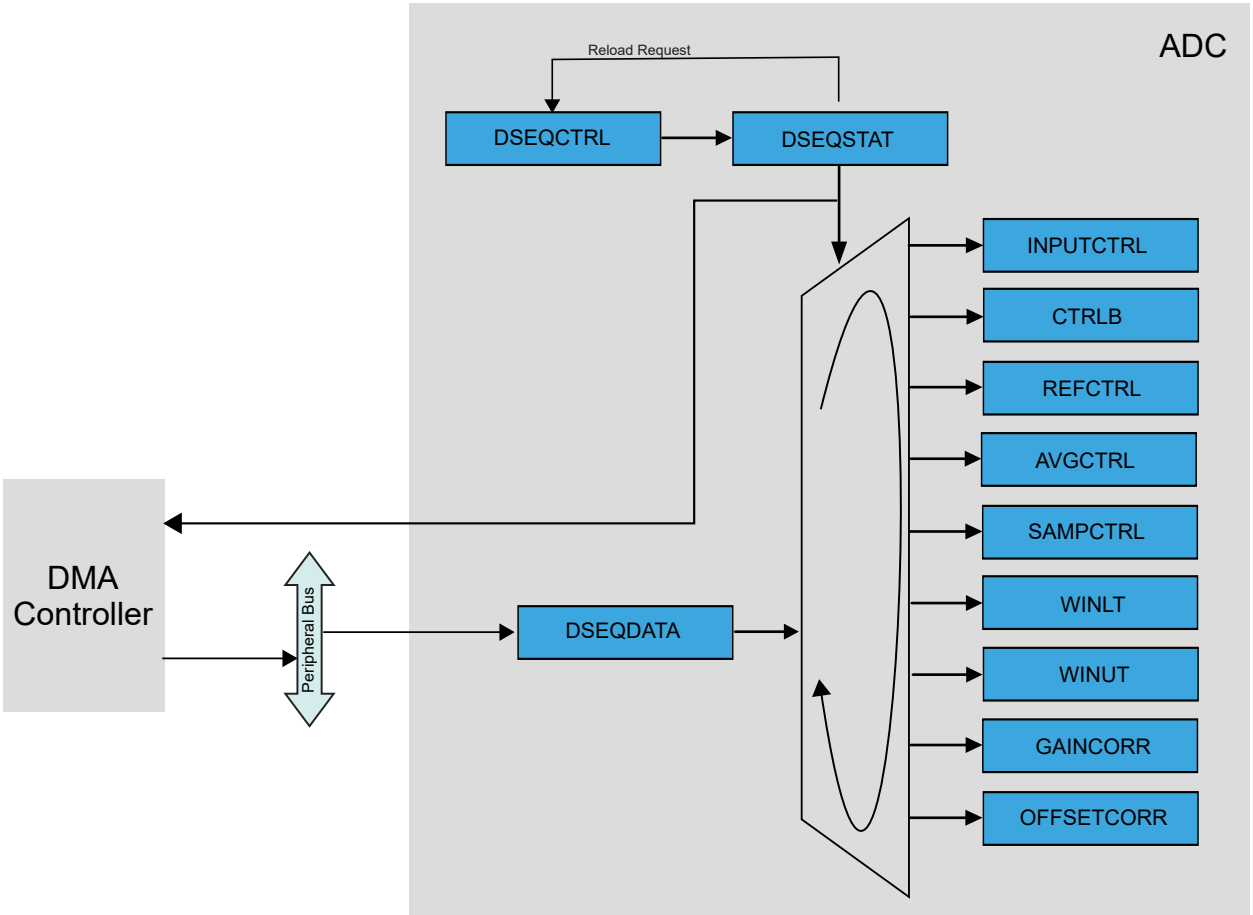

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Sampling Time Control (SAMPCTRL)
- Window Monitor Lower Threshold (WINLT)
- Window Monitor Upper Threshold (WINUT)
- Gain Correction (GAINCORR)
- Offset Correction (OFFSETCORR)

When one of these registers is written, the data is stored in the corresponding buffer as long as the current conversion is not impacted, and the corresponding busy status will be set in the Synchronization Busy register (SYNCBUSY). When a new RESULT is available, data stored in the buffer registers will be transfered to the ADC and a new conversion can start.

### 45.6.3.4. DMA Sequencing

The ADC can sequence a series of conversion. When DMA sequencing is enabled, a set of ADC configuration registers can be automatically refreshed using the DMA controller.




#### Enabling DMA Sequencing

DMA Sequencing is enabled when at least one bit in the DMA Sequence Control register (DSEQCTRL) is '1'.

When this is the case, the BUSY status bit in the DMA Sequential Status register (DSEQSTAT.BUSY) is set to '1'.

#### Disabling DMA Sequencing

DMA Sequencing is disabled when at least one of the following conditions is valid:

- The ADC is disabled (CTRLA.ENABLE = 0).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The ADC is reset (CTRLA.SWRST = 1).
- The DMA Sequence Control (DSEQCTRL) register is written '0' and the ongoing DMA sequence is completed.
- The DMA Sequencing Stop bit in the Input Control register is '1' (INPUTCTRL.DSEQSTOP = 1) and the ongoing DMA sequence is complete. One additional measurement will be done before the ADC is disabled.

When the DMA sequencing is disable, the BUSY status bit in the DMA Sequential Status register (DSEQSTAT.BUSY) is cleared and the DMA trigger generation is disabled.

If the DSEQCTRL register is written to a non-zero value, the DSEQSTOP bit in the INPUTCTRL register will be cleared and the sequencing operation will not be stopped.

**Restarting DMA Sequencing**

When the DSEQSTOP bit is set (INPUTCTRL.DSEQSTOP = 1) and the sequence is disabled (DSEQSTAT.BUSY=0), it is possible to restart the sequencing by enabling one of the following conditions:

- Write the DSEQSTOP bit in the Input Control register to zero (INPUTCTRL.DSEQSTOP = 0).
- Apply a FLUSH software command (SWTRIG.FLUSH = 1).
- Enable the flush event (EVCTRL.FLUSHEI). The sequence will restart when the flush event is received.

**DMA Sequencing Operation**

Each ADC register that is part of the DMA sequencing has a separate enable bit in the DSEQCTRL register to indicate that this field should be part of the DMA sequencing. When an enable bit in the DSEQCTRL is '1', the respective register will be updated when an access to DSEQDATA is decoded.

The DMA Sequencing (DSEQ) trigger request is generated when BUSY status bit is one (DSEQSTAT.BUSY=1), the ADC is idle or a new conversion starts, and one of the following conditions is true:

- Input Control or Control B bits in the DMA Sequential Control register is '1' (DSEQCTRL.INPUTCTRL = 1 or DSEQCTRL.CTRLB = 1)
- Reference Control, Sampling Time Control or Average Control bits in the DMA Sequential Control register is set (DSEQCTRL.REFCTRL = 1, DSEQCTRL.AVGCTRL = 1 or DSEQCTRL.SAMPCTRL = 1)
- Window Monitor Upper Threshold or Window Monitor Lower Threshold bits in the DMA Sequential Control register is set (DSEQCTRL.WINUT = 1 or DSEQCTRL.WINLT = 1)
- Offset Correction or Gain Correction bits in the DMA Sequential Control register is set (DSEQCTRL.GAINCORR = 1 or DSEQCTRL.OFFSETCORR = 1)

**Note:** When received, the DMA data must be written to the DSEQDATA register only, and only 32-bit DMA access is supported.

If a field is not enabled for DMA update, the corresponding register update will be ignored when the DSEQDATA register is written. The table below provides the DSEQ trigger generation condition and internal ADC registers refresh when the DSEQDATA register is written by the DMA.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 45-5.** DSEQ Trigger Generation and Internal ADC Register updates

| Condition | Value | Action when DMA writes to DSEQDATA |
|---|---|---|
| DSEQSTAT.INPUTCTRL or DSEQSTAT.CTRLB | 0 | • No DMA trigger is generated<br>• No data in the memory must be reserved |
| | 1 | • A DMA trigger is generated<br>• One word (32-bit) must be reserved in the memory<br>• INPUTCTRL ← DSEQDATA[15:0] if DSEQSTAT.INPUTCTRL = 1<br>• CTRLB ← DSEQDATA[31:16] if DSEQSTAT.CTRLB = 1 |
| DSEQSTAT.REFCTRL or DSEQSTAT.AVGCTRL or DSEQSTAT.SAMPCTRL | 0 | • No DMA trigger is generated<br>• No data in the memory must be reserved |
| | 1 | • A DMA trigger is generated<br>• One word (32-bit) must be reserved in the memory<br>• REFCTRL ← DSEQDATA[7:0] if DSEQSTAT.REFCTRL = 1<br>• AVGCTRL ← DSEQDATA[23:16] if DSEQSTAT.AVGCTRL = 1<br>• SAMPCTRL ← DSEQDATA[31:24] if DSEQSTAT.SAMPCTRL = 1 |
| DSEQSTAT.WINLT or DSEQSTAT.WINUT | 0 | • No DMA trigger is generated<br>• No data in the memory must be reserved |
| | 1 | • A DMA trigger is generated<br>• One word (32-bit) must be reserved in the memory<br>• WINLT ← DSEQDATA[15:0] if DSEQSTAT.WINLT = 1<br>• WINUT ← DSEQDATA[31:16] if DSEQSTAT.WINUT = 1 |
| DSEQSTAT.GAINCORR or DSEQSTAT.OFFSETCORR | 0 | • No DMA trigger is generated<br>• No data in the memory must be reserved |
| | 1 | • A DMA trigger is generated<br>• One word (32-bit) must be reserved in the memory<br>• GAINCORR ← DSEQDATA[15:0] if DSEQSTAT.GAINCORR = 1<br>• OFFSETCORR ← DSEQDATA[31:16] if DSEQSTAT.OFFSETCORR = 1 |

The DMA Sequential Status register (DSEQSTAT) stores the remaining registers to be updated by the DMA. During a sequence and when a write access to the DSEQDATA register is detected, the DSEQSTAT bits which were source of the corresponding DSEQ trigger will be cleared. When all DSEQSTAT bits are zero (except the BUSY bit), the DSEQCTRL register bits (except AUTOSTART) are copied into the DSEQSTAT register and a new DMA sequence is started when a new ADC conversion starts.

### DMA Descriptor Setup and Data Memory Organization

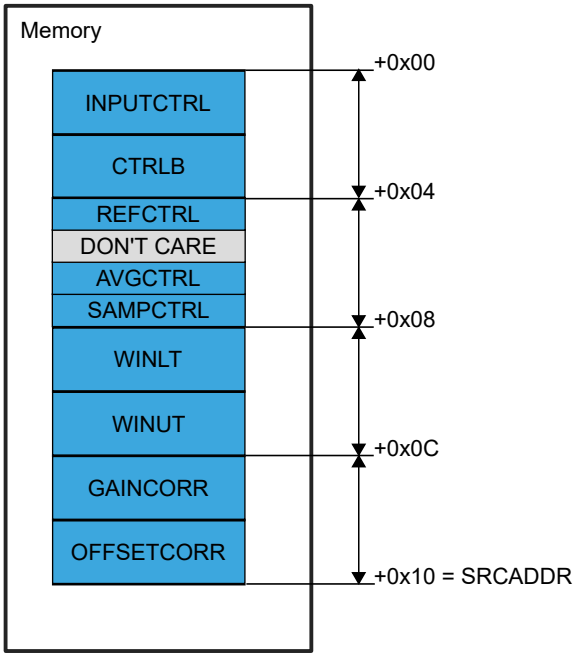
When DMA sequencing is enabled, the DMA Controller (DMAC) must be configured in the following way:

- Select 32-bit beat size transfer (DMAC.BTCTRL.BEATSIZE = WORD).
- Enable the source address increment options (DMAC.BTCTRL.SRCINC = 1, DMAC.BTCTRL.STEPSEL = SRC, DMAC.BTCTRL.STEPSIZE = X1).
- Disable the destination address increment (DMAC.BTCTRL.DSTINC = 0).
- Set the block transfer count value (DMAC.BTCNT).

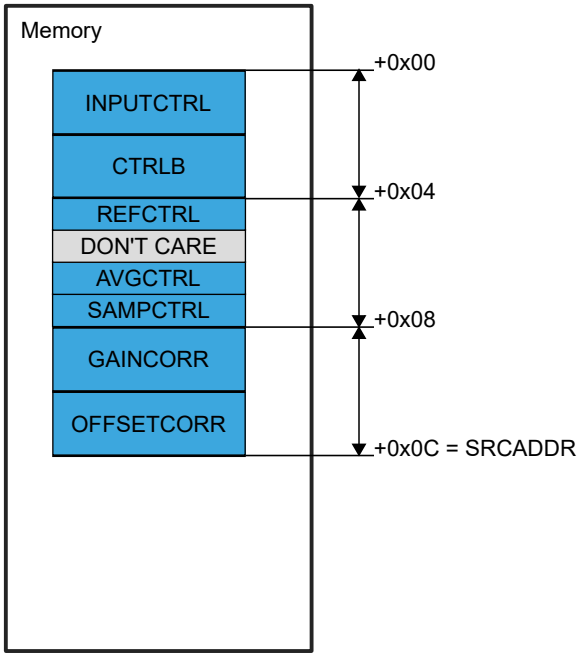

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Set the block transfer source address (DMAC.SRCADDR), as described in the DMAC Addressing section. The address corresponds to the memory section from where the DMA reads the data.
- Select the ADC.DSEQDATA address as value for the block transfer destination address (DMAC.DSTADDR = ADC.DSEQDATA address).
- Select the channel single transfer type (DMAC.CHCTRLA.BURSTLEN = SINGLE)
- Select the channel burst trigger action (DMAC.CHCTRLA.TRIGACT = BURST)
- Select the ADC DMA Sequencing trigger as channel trigger source (DMAC.CHCTRLA.TRIGSRC = DSEQ)
- Enable optional channel interrupts (DMAC.CHINTENSET)
- Enable the corresponding DMA channel (DMAC.CHCTRLA.ENABLE)

When an ADC condition is enabled to trigger a DMA transfer, one word (32-bit) will be read by the DMA from the memory source location. Since the source address is incrementing by 0x1, the data memory must be organized in a contiguous memory area. As consequence, if an ADC group of registers does not generate any DMA trigger, no data must be reserved in the memory area for this register group. The next figure shows an example of memory organization when all ADC registers are part of the sequence, and a second example where WINLT and WINUT registers are not part of the sequence.




All registers are in the sequence




WINLT / WINUT registers are not in the sequence

### Automatic Start Conversion

By default, a new conversion starts when a new start software or event trigger is received. It is also possible to automatically enable an ADC conversion by writing '1' to the AUTOSTART bit in the DSEQCTRL register (DSEQCTRL.AUTOSTART). When set, the ADC automatically starts a new conversion when a DMA sequence is complete.

**Note:** If averaging or oversampling is enabled, the new conversion automatically starts only when the previous RESULT is available (averaging or oversampling operation is complete).

**Note:** Free run mode (CTRLB.FREERUN = 1) can not be used with DMA sequencing.

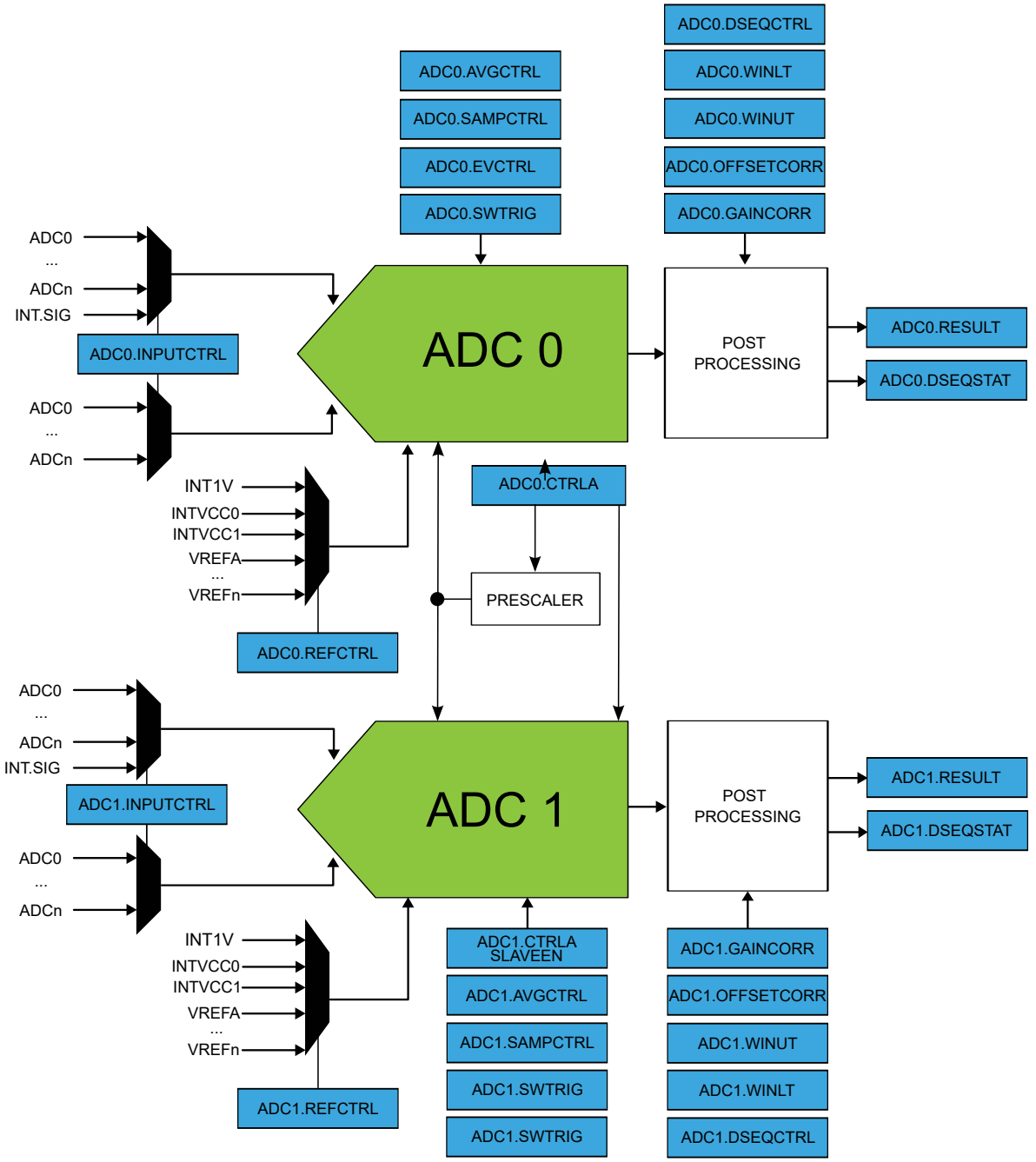
**Note:** If the conversion is triggered by event (EVCTRL.STARTEI = 1), the automatic start conversion is disabled and the AUTOSTART settings are ignored.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Addressing

### 45.6.3.5. Host - Client Operation

ADC1 will serve as a Client of ADC0 by writing a '1' to the Client Enable bit in the Control A register of the ADC1 instance (ADC1.CTRLA.SLAVEEN). When enabled, GCLK_ADC0 clock and ADC0 controls are internally routed to the ADC1 instance.




In this mode of operation, the Client ADC1 is enabled by accessing the CTRLA register of the Host ADC0. In the same way, the Host ADC event inputs will be automatically routed to the Client ADC, meaning that the input events configuration must be done in the Host ADC (ADC0.EVCTRL).

**Note:** To enable DMA sequencing for the ADC1 (Client), the ADC0 (Host) must be enabled first (ADC0.CTRLA.ENABLE = 1) before DMA sequencing is enabled (ADC1.DSEQCTRL ! = 0).

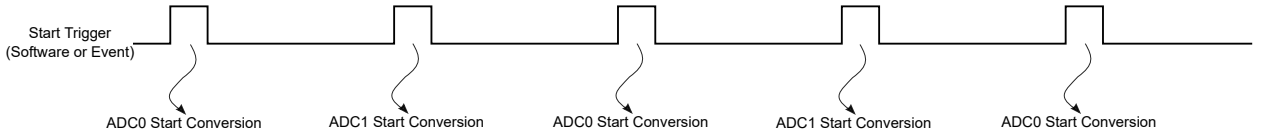

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

ADC measurements can either start simultaneously on both ADCs or be interleaved. The trigger mode selection is available in the Host ADC Control A register (ADC0.CTRLA.DUALSEL).

**Note:** The interleaved sampling is only usable in single conversion mode (ADC.CTRLB.FREERUN=0).

To restart an interleaved sequence, the user can apply different options:

- Flush the Host ADC (ADC0.SWTRIG.FLUSH = 1)
- Disable/re-enable the Host ADC (ADC0.CTRLA.ENABLE)
- Reset and reconfigure Host ADC (ADC0.CTRLA.SWRST = 1)
- Enable the flush event (EVCTRL.FLUSHEI = 1)




### 45.6.4. DMA Operation

The ADC generates the following DMA request:

- Result Conversion Ready (RESRDY): The request is set when a conversion result is available and cleared when the RESULT register is read. When the averaging operation is enabled, the DMA request is set when the averaging is completed and result is available.
- DMA Sequencing (DSEQ): For additional information, refer to the "Add link to DMA sequencing".

### 45.6.5. Interrupts

The ADC has the following interrupt sources:

- Result Conversion Ready: RESRDY
- Window Monitor: WINMON
- Overrun: OVERRUN

These interrupts, except the OVERRUN interrupt, are asynchronous wake-up sources. See *Sleep Mode Controller* for details.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the ADC is reset. See INTFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to *Nested Vector Interrupt Controller* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

INTFLAG

Interrupt Flag Status and Clear

### 45.6.6. Events

The ADC can generate the following output events:

- Result Ready (RESRDY): Generated when the conversion is complete and the result is available. Refer to EVCTRL register for details.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Window Monitor (WINMON): Generated when the window monitor condition match. Refer to CTRLB register for details.

Setting an Event Output bit in the Event Control Register (EVCTRL.xxEO=1) enables the corresponding output event. Clearing this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The ADC can take the following actions on an input event:

- Start conversion (START): Start a conversion. Refer to SWTRIG register for details.
- Conversion flush (FLUSH): Flush the conversion. Refer to SWTRIG register for details.

Setting an Event Input bit in the Event Control register (EVCTRL.xxEI=1) enables the corresponding action on input event. Clearing this bit disables the corresponding action on input event.

The ADC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the ADC will detect a rising edge on the incoming event. If the ADC action must be performed on the falling edge of the incoming event, the event line must be inverted first. This is done by setting the corresponding Event Invert Enable bit in Event Control register (EVCTRL.xINV=1).

**Note:** If several events are connected to the ADC, the enabled action will be taken on any of the incoming events. If FLUSH and START events are available at the same time, the FLUSH event has priority.

**Related Links**

EVCTRL

Event Control

CTRLB

Control B

SWTRIG

Software Trigger

Event System (EVSYS)

### 45.6.7. Sleep Mode Operation

The ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the ADC during standby sleep mode, in cases where the ADC is enabled (CTRLA.ENABLE = 1). For further details on available options, refer to Table 45-6.

**Note:** When CTRLA.ONDEMAND=1, the analog block is powered-off when the conversion is complete. When a start request is detected, the system returns from sleep and starts a new conversion after the start-up time delay.

**Table 45-6.** ADC Sleep Behavior

| CTRLA.RUNSTDBY | CTRLA.ONDEMAND | CTRLA.ENABLE | Description |
|---|---|---|---|
| x | x | 0 | Disabled |
| 0 | 0 | 1 | Run in all sleep modes except STANDBY. |
| 0 | 1 | 1 | Run in all sleep modes on request, except STANDBY. |
| 1 | 0 | 1 | Run in all sleep modes. |
| 1 | 1 | 1 | Run in all sleep modes on request. |

### 45.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The Software Reset bit in the Control A register (CTRLA.SWRST)
- The Enable bit in the Control A register (CTRLA.ENABLE)

The following registers are synchronized when written:

- The Input Control register (INPUTCTRL)
- The Control B register (CTRLB)
- The Reference Control (REFCTRL)
- The Average control register (AVGCTRL)
- The Sampling time control register (SAMPCTRL)
- The Window Monitor Lower Threshold register (WINLT)
- The Window Monitor Upper Threshold register (WINUT)
- The Gain correction register (GAINCORR)
- The Offset Correction register (OFFSETCORR)
- The Software Trigger register (SWTRIG)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 45.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ONDEMAND | RUNSTDBY | SLAVEEN | DUALSEL[1:0] | | | ENABLE | SWRST |
| | | 15:8 | R2R | | | | | PRESCALER[2:0] | | |
| 0x02 | EVCTRL | 7:0 | | | WINMONEO | RESRDYEO | STARTINV | FLUSHINV | STARTEI | FLUSHEI |
| 0x03 | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x04 | INPUTCTRL | 7:0 | DIFFMODE | | | MUXPOS[4:0] | | | | |
| | | 15:8 | DSEQSTOP | | | MUXNEG[4:0] | | | | |
| 0x06 | CTRLB | 7:0 | | | | RESSEL[1:0] | | CORREN | FREERUN | LEFTADJ |
| | | 15:8 | | | | | WINSS | WINMODE[2:0] | | |
| 0x08 | REFCTRL | 7:0 | REFCOMP | | | | REFSEL[3:0] | | | |
| 0x09 | Reserved | | | | | | | | | |
| 0x0A | AVGCTRL | 7:0 | | | ADJRES[2:0] | | SAMPLENUM[3:0] | | | |
| 0x0B | SAMPCTRL | 7:0 | OFFCOMP | | SAMPLEN[5:0] | | | | | |
| 0x0C | WINLT | 7:0 | WINLT[7:0] | | | | | | | |
| | | 15:8 | WINLT[15:8] | | | | | | | |
| 0x0E | WINUT | 7:0 | WINUT[7:0] | | | | | | | |
| | | 15:8 | WINUT[15:8] | | | | | | | |
| 0x10 | GAINCORR | 7:0 | GAINCORR[7:0] | | | | | | | |
| | | 15:8 | | | | | GAINCORR[11:8] | | | |
| 0x12 | OFFSETCORR | 7:0 | OFFSETCORR[7:0] | | | | | | | |
| | | 15:8 | | | | | OFFSETCORR[11:8] | | | |
| 0x14 | SWTRIG | 7:0 | | | | | | | START | FLUSH |
| 0x15<br>...<br>0x2B | Reserved | | | | | | | | | |
| 0x2C | INTENCLR | 7:0 | | | | | | WINMON | OVERRUN | RESRDY |
| 0x2D | INTENSET | 7:0 | | | | | | WINMON | OVERRUN | RESRDY |
| 0x2E | INTFLAG | 7:0 | | | | | | WINMON | OVERRUN | RESRDY |
| 0x2F | STATUS | 7:0 | WCC[5:0] | | | | | | | ADCBUSY |
| 0x30 | SYNCBUSY | 7:0 | WINLT | SAMPCTRL | AVGCTRL | REFCTRL | CTRLB | INPUTCTRL | ENABLE | SWRST |
| | | 15:8 | | | | | SWTRIG | OFFSETCORR | GAINCORR | WINUT |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | DSEQDATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x38 | DSEQCTRL | 7:0 | GAINCORR | WINUT | WINLT | SAMPCTRL | AVGCTRL | REFCTRL | CTRLB | INPUTCTRL |
| | | 15:8 | | | | | | | | OFFSETCORR |
| | | 23:16 | | | | | | | | |
| | | 31:24 | AUTOSTART | | | | | | | |
| 0x3C | DSEQSTAT | 7:0 | GAINCORR | WINUT | WINLT | SAMPCTRL | AVGCTRL | REFCTRL | CTRLB | INPUTCTRL |
| | | 15:8 | | | | | | | | OFFSETCORR |
| | | 23:16 | | | | | | | | |
| | | 31:24 | BUSY | | | | | | | |
| 0x40 | RESULT | 7:0 | RESULT[7:0] | | | | | | | |
| | | 15:8 | RESULT[15:8] | | | | | | | |
| 0x42<br>...<br>0x43 | Reserved | | | | | | | | | |
| 0x44 | RESS | 7:0 | RESS[7:0] | | | | | | | |
| | | 15:8 | RESS[15:8] | | | | | | | |
| 0x46<br>...<br>0x47 | Reserved | | | | | | | | | |
| 0x48 | CALIB | 7:0 | | | BIASR2R[2:0] | | | BIASCOMP[2:0] | | |
| | | 15:8 | | | | | | BIASREFBUF[2:0] | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 45.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to the section on Synchronization.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization section.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

**Related Links**

Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x0000
**Property:** Enable-Protected, PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | R2R | | | | | PRESCALER[2:0] | | |
| Access | R/W | | | | | R/W | R/W | R/W |
| Reset | 0 | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | SLAVEEN | DUALSEL[1:0] | | | ENABLE | SWRST |
| Access | R/W | R/W | R/W | R/W | R/W | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |

**Bit 15 – R2R** Rail to Rail Operation Enable

| Value | Description |
|---|---|
| 0 | Rail-to-Rail operation disable |
| 1 | Rail-to-Rail operation enable. The R2R bit must be set to '1' only in differential mode. |

**Bits 10:8 – PRESCALER[2:0]** Prescaler Configuration

This field defines the ADC clock relative to the peripheral clock according Table below. This field is not synchronized. For the Client ADC, these bits have no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV2 | Peripheral clock divided by 2 |
| 0x1 | DIV4 | Peripheral clock divided by 4 |
| 0x2 | DIV8 | Peripheral clock divided by 8 |
| 0x3 | DIV16 | Peripheral clock divided by 16 |
| 0x4 | DIV32 | Peripheral clock divided by 32 |
| 0x5 | DIV64 | Peripheral clock divided by 64 |
| 0x6 | DIV128 | Peripheral clock divided by 128 |
| 0x7 | DIV256 | Peripheral clock divided by 256 |

**Bit 7 – ONDEMAND** On Demand Control

The On Demand operation mode allows the ADC to be enabled or disabled, depending on other peripheral requests.
In On Demand operation mode, i.e., if the ONDEMAND bit has been previously set, the ADC will only be running when requested by a peripheral. If there is no peripheral requesting the ADC will be in a disable state.
If On Demand is disabled the ADC will always be running when enabled.
In standby sleep mode, the On Demand operation is still active if the CTRLA.RUNSTDBY bit is '1'. If CTRLA.RUNSTDBY is '0', the ADC is disabled.
This bit is not synchronized.
**Note:** For the Client ADC, this bit has no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1). ONDEMAND bit from Host ADC instance will control the On Demand operation mode.

| Value | Description |
|---|---|
| 0 | The ADC is always on , if enabled. |
| 1 | The ADC is enabled, when a peripheral is requesting the ADC conversion. The ADC is disabled if no peripheral is requesting it. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 6 – RUNSTDBY** Run in Standby

This bit controls how the ADC behaves during standby sleep mode.
This bit is not synchronized.
**Note:** For the Client ADC, this bit has no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1). RUNSTDBY bit from Host ADC instance will control the Client ADC operation in standby sleep mode.

| Value | Description |
|---|---|
| 0 | The ADC is halted during standby sleep mode. |
| 1 | The ADC is not stopped in standby sleep mode. If CTRLA.ONDEMAND=1, the ADC will be running when a peripheral is requesting it. If CTRLA.ONDEMAND=0, the ADC will always be running in standby sleep mode. |

**Bit 5 – SLAVEEN** Client Enable

This bit enables the Host/Client operation and it is available only in the Client ADC instance.
This bit is not synchronized and can be set only for the Client ADC. For the Host ADC, this bit is always read zero.

| Value | Description |
|---|---|
| 0 | The Host/Client operation is disabled |
| 1 | The ADC1 is enabled as a Client of ADC0 |

**Bits 4:3 – DUALSEL[1:0]** Dual Mode Trigger Selection

These bits define the trigger mode, as shown in Table below. These bits are available in the Host ADC and have no effect if the Host/Client operation is disabled (ADC1.CTRLA.SLAVEEN=0).

| Value | Name | Description |
|---|---|---|
| 0x0 | BOTH | Start event or software trigger will start a conversion on both ADCs |
| 0x1 | INTERLEAVE | START event or software trigger will alternatingly start a conversion on ADC0 and ADC1.<br>**Note:** The interleaved sampling is only usable in single conversion mode (ADC.CTRLB.FREERUN=0). |
| 0x2 – 0x3 | | Reserved |

**Bit 1 – ENABLE** Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the ENABLE bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
For the Client ADC, this bit has no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

| Value | Description |
|---|---|
| 0 | The ADC is disabled. |
| 1 | The ADC is enabled. |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the ADC, except DBGCTRL, to their initial state, and the ADC will be disabled.
Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.2. Event Control

**Name:** EVCTRL
**Offset:** 0x02
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | WINMONEO | RESRDYEO | STARTINV | FLUSHINV | STARTEI | FLUSHEI |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 5 – WINMONEO** Window Monitor Event Out
This bit indicates whether the Window Monitor event output is enabled or not and an output event will be generated when the window monitor detects something.

| Value | Description |
|---|---|
| 0 | Window Monitor event output is disabled and an event will not be generated. |
| 1 | Window Monitor event output is enabled and an event will be generated. |

**Bit 4 – RESRDYEO** Result Ready Event Out
This bit indicates whether the Result Ready event output is enabled or not and an output event will be generated when the conversion result is available.

| Value | Description |
|---|---|
| 0 | Result Ready event output is disabled and an event will not be generated. |
| 1 | Result Ready event output is enabled and an event will be generated. |

**Bit 3 – STARTINV** Start Conversion Event Invert Enable
For the Client ADC, this bit has no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

| Value | Description |
|---|---|
| 0 | Start event input source is not inverted. |
| 1 | Start event input source is inverted. |

**Bit 2 – FLUSHINV** Flush Event Invert Enable
For the Client ADC, this bit has no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

| Value | Description |
|---|---|
| 0 | Flush event input source is not inverted. |
| 1 | Flush event input source is inverted. |

**Bit 1 – STARTEI** Start Conversion Event Input Enable
For the Client ADC, this bit has no effect when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

| Value | Description |
|---|---|
| 0 | A new conversion will not be triggered on any incoming event. |
| 1 | A new conversion will be triggered on any incoming event. |

**Bit 0 – FLUSHEI** Flush Event Input Enable
For a Client ADC, this bit has no effect when the respective SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

| Value | Description |
|---|---|
| 0 | A flush and new conversion will not be triggered on any incoming event. |
| 1 | A flush and new conversion will be triggered on any incoming event. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.3. Debug Control

**Name:** DBGCTRL
**Offset:** 0x03
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
This bit is not reset by a software reset.
This bit controls the functionality when the CPU is halted by an external debugger.
This bit should be written only while a conversion is not ongoing.
When Client operation is enabled, Host and Client ADC instances must have the same DBGRUN bit value to ensure proper operation.

| Value | Description |
|---|---|
| 0 | The ADC is halted when the CPU is halted by an external debugger. |
| 1 | The ADC continues normal operation when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.4. Input Control

**Name:** INPUTCTRL
**Offset:** 0x04
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DSEQSTOP | | | MUXNEG[4:0] | | | | |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIFFMODE | | | MUXPOS[4:0] | | | | |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 15 – DSEQSTOP** Stop DMA Sequencing
When the bit is set, the DMA sequencing automatically stops when the last sequence configuration is complete.
**Note:** one more conversion will be done after the last sequence is complete.

**Bits 12:8 – MUXNEG[4:0]** Negative MUX Input Selection
These bits define the MUX selection for the negative ADC input.

| Value | Name | Description |
|---|---|---|
| 0x00 | AIN0 | ADC AIN0 pin |
| 0x01 | AIN1 | ADC AIN1 pin |
| 0x02 | AIN2 | ADC AIN2 pin |
| 0x03 | AIN3 | ADC AIN3 pin |
| 0x04 | AIN4 | ADC AIN4 pin |
| 0x05 | AIN5 | ADC AIN5 pin |
| 0x06 | AIN6 | ADC AIN6 pin |
| 0x07 | AIN7 | ADC AIN7 pin |
| 0x08 - 0x17 | | Reserved |
| 0x18 | GND | Internal ground |
| 0x19 - 0x1F | | Reserved |

**Bit 7 – DIFFMODE** Differential Mode

| Value | Description |
|---|---|
| 0x0 | The ADC is running in singled-ended mode. |
| 0x1 | The ADC is running in differential mode. In this mode, the voltage difference between the MUXPOS and MUXNEG inputs will be converted by the ADC. |

**Bits 4:0 – MUXPOS[4:0]** Positive MUX Input Selection
These bits define the MUX selection for the positive ADC input. If the internal bandgap voltage or temperature sensor input channel is selected, then the Sampling Time Length bit group in the Sampling Control register must be written with a corresponding value, as shown in "Table 54-24. Operating Conditions".

| Value | Name | Description |
|---|---|---|
| 0x00 | AIN0 | ADC AIN0 pin |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x01 | AIN1 | ADC AIN1 pin |
| 0x02 | AIN2 | ADC AIN2 pin |
| 0x03 | AIN3 | ADC AIN3 pin |
| 0x04 | AIN4 | ADC AIN4 pin |
| 0x05 | AIN5 | ADC AIN5 pin |
| 0x06 | AIN6 | ADC AIN6 pin |
| 0x07 | AIN7 | ADC AIN7 pin |
| 0x08 | AIN8 | ADC AIN8 pin |
| 0x09 | AIN9 | ADC AIN9 pin |
| 0x0A | AIN10 | ADC AIN10 pin |
| 0x0B | AIN11 | ADC AIN11 pin |
| 0x0C | AIN12 | ADC AIN12 pin |
| 0x0D | AIN13 | ADC AIN13 pin |
| 0x0E | AIN14 | ADC AIN14 pin |
| 0x0F | AIN15 | ADC AIN15 pin |
| 0x10 – 0x17 | - | Reserved |
| 0x18 | SCALEDCOREVCC | 1/4 Scaled Core Supply |
| 0x19 | SCALEDVBAT | 1/4 Scaled VBAT Supply |
| 0x1A | SCALEDIOVCC | 1/4 Scaled I/O Supply |
| 0x1B | BANDGAP | Bandgap Voltage |
| 0x1C | PTAT | Temperature Sensor |
| 0x1D | CTAT | Temperature Sensor |
| 0x1E | DAC0 | DAC0 Output |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.5. Control B

**Name:** CTRLB
**Offset:** 0x06
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | WINSS | WINMODE[2:0] | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | RESSEL[1:0] | | CORREN | FREERUN | LEFTADJ |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bit 11 – WINSS** Window Single Sample

When this bit is written the window functionality is working on each conversions and not on the accumulated value. The number of convesions matching with the window comparator is available on STATUS register (STATUS.WCC). The last sample result is available on RESS register.

**Bits 10:8 – WINMODE[2:0]** Window Monitor Mode

These bits enable and define the window monitor mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | No window mode (default) |
| 0x1 | MODE1 | RESULT > WINLT |
| 0x2 | MODE2 | RESULT < WINUT |
| 0x3 | MODE3 | WINLT < RESULT < WINUT |
| 0x4 | MODE4 | !(WINLT < RESULT < WINUT) |
| 0x5 – 0x7 | | Reserved |

**Bits 4:3 – RESSEL[1:0]** Conversion Result Resolution

These bits define whether the ADC completes the conversion 12-, 10- or 8-bit result resolution.

| Value | Name | Description |
|---|---|---|
| 0x0 | 12BIT | 12-bit result |
| 0x1 | 16BIT | For averaging mode output |
| 0x2 | 10BIT | 10-bit result |
| 0x3 | 8BIT | 8-bit result |

**Bit 2 – CORREN** Digital Correction Logic Enable

The ADC conversion result in the RESULT register is then corrected for gain and offset based on the values in the GAINCORR and OFFSETCORR registers. Conversion time will be increased by 13 cycles according to the value in the Offset Correction Value bit group in the Offset Correction register.

| Value | Description |
|---|---|
| 0 | Disable the digital result correction |
| 1 | Enable the digital result correction |

**Bit 1 – FREERUN** Free Running Mode

| Value | Description |
|---|---|
| 0 | The ADC run in single conversion mode |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | The ADC is in free running mode and a new conversion will be initiated when a previous conversion completes |

**Bit 0 – LEFTADJ** Left-Adjusted Result

The high byte of the 12-bit result will be present in the upper part of the result register. Writing this bit to zero (default) will right-adjust the value in the RESULT register.

| Value | Description |
|---|---|
| 0 | The ADC conversion result is right-adjusted in the RESULT register |
| 1 | The ADC conversion result is left-adjusted in the RESULT register |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 45.8.6. Reference Control

**Name:** REFCTRL
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | REFCOMP | | | | REFSEL[3:0] | | | |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 7 – REFCOMP** Reference Buffer Offset Compensation Enable
The gain error can be reduced by enabling the reference buffer offset compensation. This will increase the start-up time of the reference.

| Value | Description |
|---|---|
| 0 | Reference buffer offset compensation is disabled. |
| 1 | Reference buffer offset compensation is enabled. |

**Bits 3:0 – REFSEL[3:0]** Reference Selection
These bits select the reference for the ADC.

| Value | Name | Description |
|---|---|---|
| 0x0 | INTREF | internal bandgap reference, refer to the VREF.SEL bit-field in SUPC-Supply Controller for more details |
| x01 | | Reserved |
| 0x2 | INTVCC0 | 1/2 VDDANA (only for VDDANA > 2.0v) |
| 0x3 | INTVCC1 | VDDANA |
| 0x4 | AREFA | External reference |
| 0x5 | AREFB | External reference |
| 0x6 | AREFC | External reference (ADC1 only) |
| other | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.7. Average Control

**Name:** AVGCTRL
**Offset:** 0x0A
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | ADJRES[2:0] | | | SAMPLENUM[3:0] | | | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 6:4 – ADJRES[2:0]** Adjusting Result / Division Coefficient
These bits define the division coefficient in 2^n steps.

**Bits 3:0 – SAMPLENUM[3:0]** Number of Samples to be Collected
These bits define how many samples are added together. The result will be available in the Result register (RESULT). Note: if the result width increases, CTRLB.RESSEL must be changed.

| Value | Description |
|---|---|
| 0x0 | 1 sample |
| 0x1 | 2 samples |
| 0x2 | 4 samples |
| 0x3 | 8 samples |
| 0x4 | 16 samples |
| 0x5 | 32 samples |
| 0x6 | 64 samples |
| 0x7 | 128 samples |
| 0x8 | 256 samples |
| 0x9 | 512 samples |
| 0xA | 1024 samples |
| 0xB - 0xF | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.8. Sampling Time Control

**Name:** SAMPCTRL
**Offset:** 0x0B
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OFFCOMP | | SAMPLEN[5:0] | | | | | |
| Access | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – OFFCOMP** Comparator Offset Compensation Enable
Setting this bit enables the offset compensation for each sampling period to ensure low offset and immunity to temperature or voltage drift. This compensation increases the sampling time by three clock cycles that results in a fixed sampling duration of 4 CLK_ADC cycles.
This bit must be set to zero to validate the SAMPLEN value. It's not possible to use OFFCOMP=1 and SAMPLEN>0.

**Bits 5:0 – SAMPLEN[5:0]** Sampling Time Length
These bits control the ADC sampling time in number of CLK_ADC cycles, depending of the prescaler value, thus controlling the ADC input impedance. Sampling time is set according to the equation:
$\text{Sampling time} = (\text{SAMPLEN}+1) \cdot (\text{CLK}_{\text{ADC}})$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.9. Window Monitor Lower Threshold

**Name:** WINLT
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WINLT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WINLT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – WINLT[15:0]** Window Lower Threshold
If the window monitor is enabled, these bits define the lower threshold value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.10. Window Monitor Upper Threshold

**Name:** WINUT
**Offset:** 0x0E
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WINUT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WINUT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – WINUT[15:0]** Window Upper Threshold
If the window monitor is enabled, these bits define the upper threshold value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.11. Gain Correction

**Name:** GAINCORR
**Offset:** 0x10
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | GAINCORR[11:8] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GAINCORR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 11:0 – GAINCORR[11:0]** Gain Correction Value
If CTRLB.CORREN=1, these bits define how the ADC conversion result is compensated for gain error before being written to the result register. The gain correction is a fractional value, a 1-bit integer plus an 11-bit fraction, and therefore ½ <= GAINCORR < 2. GAINCORR values range from 0.10000000000 to 1.11111111111.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.12. Offset Correction

**Name:** OFFSETCORR
**Offset:** 0x12
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | OFFSETCORR[11:8] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OFFSETCORR[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 11:0 – OFFSETCORR[11:0]** Offset Correction Value
If CTRLB.CORREN=1, these bits define how the ADC conversion result is compensated for offset error before being written to the Result register. This OFFSETCORR value is in two's complement format.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 45.8.13. Software Trigger

**Name:** SWTRIG
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | START | FLUSH |
| Access | | | | | | | W | RW |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – START** Start ADC Conversion
Writing a '1' to this bit will start a conversion or sequence. The bit is cleared by hardware when the conversion has started. Writing a '1' to this bit when it is already set has no effect.
Writing a '0' to this bit has no effect.

**Bit 0 – FLUSH** ADC Conversion Flush
Writing a '1' to this bit will flush the ADC pipeline. A flush will restart the ADC clock on the next peripheral clock edge, and all conversions in progress will be aborted and lost. This bit will be cleared after the ADC has been flushed.
After the flush, the ADC will resume where it left off; i.e., if a conversion was pending, the ADC will start a new conversion.
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.14. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x2C
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WINMON | OVERRUN | RESRDY |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – WINMON** Window Monitor Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Window Monitor Interrupt Enable bit, which disables the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The window monitor interrupt is disabled. |
| 1 | The window monitor interrupt is enabled, and an interrupt request will be generated when the Window Monitor interrupt flag is set. |

**Bit 1 – OVERRUN** Overrun Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Interrupt Enable bit, which disables the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Overrun interrupt is disabled. |
| 1 | The Overrun interrupt is enabled, and an interrupt request will be generated when the Overrun interrupt flag is set. |

**Bit 0 – RESRDY** Result Ready Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Result Ready Interrupt Enable bit, which disables the corresponding interrupt request.

| Value | Description |
|---|---|
| 0 | The Result Ready interrupt is disabled. |
| 1 | The Result Ready interrupt is enabled, and an interrupt request will be generated when the Result Ready interrupt flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.15. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x2D
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WINMON | OVERRUN | RESRDY |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – WINMON** Window Monitor Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Window Monitor Interrupt bit, which enables the Window Monitor interrupt.

| Value | Description |
|---|---|
| 0 | The Window Monitor interrupt is disabled. |
| 1 | The Window Monitor interrupt is enabled. |

**Bit 1 – OVERRUN** Overrun Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Interrupt bit, which enables the Overrun interrupt.

| Value | Description |
|---|---|
| 0 | The Overrun interrupt is disabled. |
| 1 | The Overrun interrupt is enabled. |

**Bit 0 – RESRDY** Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Result Ready Interrupt bit, which enables the Result Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Result Ready interrupt is disabled. |
| 1 | The Result Ready interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.16. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x2E
**Reset:** 0x00
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WINMON | OVERRUN | RESRDY |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – WINMON** Window Monitor Interrupt Flag
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.
This flag is set on the next GCLK_ADC cycle after a match with the window monitor condition, and an interrupt request will be generated if INTENCLR/SET.WINMON is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window Monitor interrupt flag.

**Bit 1 – OVERRUN** Overrun Interrupt Flag
This flag is cleared by writing a '1' to the flag.
This flag is set if RESULT is written before the previous value has been read by CPU, and an interrupt request will be generated if INTENCLR/SET.OVERRUN=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overrun interrupt flag.

**Bit 0 – RESRDY** Result Ready Interrupt Flag
This flag is cleared by writing a '1' to the flag or by reading the RESULT register.
This flag is set when the conversion result is available, and an interrupt will be generated if INTENCLR/SET.RESRDY=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Result Ready interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.17. STATUS

**Name:** STATUS
**Offset:** 0x2F
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WCC[5:0] | | | | | | | ADCBUSY |
| Access | R | R | R | R | R | R | | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | 0 |

**Bits 7:2 – WCC[5:0]** Window Comparator Counter
These bits indicates the number of sample matching with the window comparator.
Writing a zero to this bit will have no effect.
Writing a one to this bit will have no effect.

**Bit 0 – ADCBUSY** ADC Busy Status
This bit is read one when the data acquisition in on going.
Writing a zero to this bit will have no effect.
Writing a one to this bit will have no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.18. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SWTRIG | OFFSETCORR | GAINCORR | WINUT |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WINLT | SAMPCTRL | AVGCTRL | REFCTRL | CTRLB | INPUTCTRL | ENABLE | SWRST |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 11 – SWTRIG** Software Trigger Synchronization Busy
This bit is cleared when the synchronization of SWTRIG register between the clock domains is complete.
This bit is set when the synchronization of SWTRIG register between clock domains is started.
**Note:** For the Client ADC, this bit is always read zero when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

**Bit 10 – OFFSETCORR** Offset Correction Synchronization Busy
This bit is cleared when the synchronization of OFFSETCORR register between the clock domains is complete.
This bit is set when the synchronization of OFFSETCORR register between clock domains is started.

**Bit 9 – GAINCORR** Gain Correction Synchronization Busy
This bit is cleared when the synchronization of GAINCORR register between the clock domains is complete.
This bit is set when the synchronization of GAINCORR register between clock domains is started.

**Bit 8 – WINUT** Window Monitor Upper Threshold Synchronization Busy
This bit is cleared when the synchronization of WINUT register between the clock domains is complete.
This bit is set when the synchronization of WINUT register between clock domains is started.

**Bit 7 – WINLT** Window Monitor Lower Threshold Synchronization Busy
This bit is cleared when the synchronization of WINLT register between the clock domains is complete.
This bit is set when the synchronization of WINLT register between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 6 – SAMPCTRL** Sampling Time Control Synchronization Busy
This bit is cleared when the synchronization of SAMPCTRL register between the clock domains is complete.
This bit is set when the synchronization of SAMPCTRL register between clock domains is started.

**Bit 5 – AVGCTRL** Average Control Synchronization Busy
This bit is cleared when the synchronization of AVGCTRL register between the clock domains is complete.
This bit is set when the synchronization of AVGCTRL register between clock domains is started.

**Bit 4 – REFCTRL** Reference Control Synchronization Busy
This bit is cleared when the synchronization of REFCTRL register between the clock domains is complete.
This bit is set when the synchronization of REFCTRL register between clock domains is started.

**Bit 3 – CTRLB** Control B Synchronization Busy
This bit is cleared when the synchronization of CTRLB register between the clock domains is complete.
This bit is set when the synchronization of CTRLB register between clock domains is started.

**Bit 2 – INPUTCTRL** Input Control Synchronization Busy
This bit is cleared when the synchronization of INPUTCTRL register between the clock domains is complete.
This bit is set when the synchronization of INPUTCTRL register between clock domains is started.

**Bit 1 – ENABLE** ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE register between the clock domains is complete.
This bit is set when the synchronization of ENABLE register between clock domains is started.
**Note:** For the Client ADC, this bit is always read zero when the SLAVEEN bit is set (CTRLA.SLAVEEN= 1).

**Bit 0 – SWRST** SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST register between the clock domains is complete.
This bit is set when the synchronization of SWRST register between clock domains is started


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.19. DSEQDATA

**Name:** DSEQDATA
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** DMA Sequential Data
This register stores data written by the DMA and re-directed to the first enabled ADC registers in the DSEQSTAT register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.20. DSEQCTRL

**Name:** DSEQCTRL
**Offset:** 0x38
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | AUTOSTART | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | OFFSETCORR |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GAINCORR | WINUT | WINLT | SAMPCTRL | AVGCTRL | REFCTRL | CTRLB | INPUTCTRL |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – AUTOSTART** ADC Auto-Start Conversion

| Value | Description |
|---|---|
| 0 | ADC conversion starts when a DMA sequence is complete and a start software or event trigger is received. |
| 1 | ADC conversion automatically starts when a DMA sequence is complete. This setting is ignored if the convertion start by event is enabled (EVCTRL.STARTEI=1). |

**Bit 8 – OFFSETCORR** Offset Correction

| Value | Description |
|---|---|
| 0 | DMA update of the Offset Correction register is disabled. |
| 1 | DMA update of the Offset Correction register is enabled. |

**Bit 7 – GAINCORR** Gain Correction

| Value | Description |
|---|---|
| 0 | DMA update of the Gain Correction register is disabled. |
| 1 | DMA update of the Gain Correction register is enabled. |

**Bit 6 – WINUT** Window Monitor Upper Threshold

| Value | Description |
|---|---|
| 0 | DMA update of the Window Monitor Upper Threshold register is disabled. |
| 1 | DMA update of the Window Monitor Upper Threshold register is enabled. |

**Bit 5 – WINLT** Window Monitor Lower Threshold

| Value | Description |
|---|---|
| 0 | DMA update of the Window Monitor Lower Threshold register is disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | DMA update of the Window Monitor Lower Threshold register is enabled. |

**Bit 4 – SAMPCTRL** Sampling Time Control

| Value | Description |
|---|---|
| 0 | DMA update of the Sampling Time Control register is disabled. |
| 1 | DMA update of the Sampling Time Control register is enabled. |

**Bit 3 – AVGCTRL** Average Control

| Value | Description |
|---|---|
| 0 | DMA update of the Average Control register is disabled. |
| 1 | DMA update of the Average Control register is enabled. |

**Bit 2 – REFCTRL** Reference Control

| Value | Description |
|---|---|
| 0 | DMA update of the Reference Control register is disabled. |
| 1 | DMA update of the Reference Control register is enabled. |

**Bit 1 – CTRLB** Control B

| Value | Description |
|---|---|
| 0 | DMA update of the Control B register is disabled. |
| 1 | DMA update of the Control B register is enabled. |

**Bit 0 – INPUTCTRL** Input Control

| Value | Description |
|---|---|
| 0 | DMA update of the Input Control register is disabled. |
| 1 | DMA update of the Input Control register is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 45.8.21. DSEQSTAT

**Name:** DSEQSTAT
**Offset:** 0x3C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BUSY | | | | | | | |
| Access | R | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | OFFSETCORR |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GAINCORR | WINUT | WINLT | SAMPCTRL | AVGCTRL | REFCTRL | CTRLB | INPUTCTRL |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – BUSY** DMA Sequencing Busy
The bit is set when the DMA sequencing is enabled or restarted.
The bit is cleared when the DMA sequencing is disabled.

**Bit 8 – OFFSETCORR** Offset Correction

| Value | Description |
|---|---|
| 0 | DMA update of the Offset Correction register is complete or disabled. |
| 1 | DMA update of the Offset Correction register is enabled. |

**Bit 7 – GAINCORR** Gain Correction

| Value | Description |
|---|---|
| 0 | DMA update of the Gain Correction register is complete or disabled. |
| 1 | DMA update of the Gain Correction register is enabled. |

**Bit 6 – WINUT** Window Monitor Upper Threshold

| Value | Description |
|---|---|
| 0 | DMA update of the Window Monitor Upper Threshold register is complete or disabled. |
| 1 | DMA update of the Window Monitor Upper Threshold register is enabled. |

**Bit 5 – WINLT** Window Monitor Lower Threshold

| Value | Description |
|---|---|
| 0 | DMA update of the Window Monitor Lower Threshold register is complete or disabled. |
| 1 | DMA update of the Window Monitor Lower Threshold register is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 4 – SAMPCTRL** Sampling Time Control

| Value | Description |
|---|---|
| 0 | DMA update of the Sampling Time Control register is complete or disabled. |
| 1 | DMA update of the Sampling Time Control register is enabled. |

**Bit 3 – AVGCTRL** Average Control

| Value | Description |
|---|---|
| 0 | DMA update of the Average Control register is complete or disabled. |
| 1 | DMA update of the Average Control register is enabled. |

**Bit 2 – REFCTRL** Reference Control

| Value | Description |
|---|---|
| 0 | DMA update of the Reference Control register is complete or disabled. |
| 1 | DMA update of the Reference Control register is enabled. |

**Bit 1 – CTRLB** Control B

| Value | Description |
|---|---|
| 0 | DMA update of the Control B register is complete or disabled. |
| 1 | DMA update of the Control B register is enabled. |

**Bit 0 – INPUTCTRL** Input Control

| Value | Description |
|---|---|
| 0 | DMA update of the Input Control register is complete or disabled. |
| 1 | DMA update of the Input Control register is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.22. Result

**Name:** RESULT
**Offset:** 0x40
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RESULT[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RESULT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RESULT[15:0]** Result Conversion Value

These bits will hold up to a 16-bit ADC conversion result, depending on the configuration.
In single conversion mode without averaging, the ADC conversion will produce a 12-bit result, which can be left- or right-shifted, depending on the setting of CTRLB.LEFTADJ.
If the result is left-adjusted (CTRLB.LEFTADJ), the high byte of the result will be in bit position [15:8], while the remaining 4 bits of the result will be placed in bit locations [7:4]. This can be used only if an 8-bit result is needed; i.e., one can read only the high byte of the entire 16-bit register.
If the result is not left-adjusted (CTRLB.LEFTADJ) and no oversampling is used, the result will be available in bit locations [11:0], and the result is then 12 bits long. If oversampling is used, the result will be located in bit locations [15:0], depending on the settings of the Average Control register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.23. RESS

**Name:** RESS
**Offset:** 0x44
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RESS[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RESS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RESS[15:0]** Last ADC Conversion Result
These bits will hold up the last ADC conversion result.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 45.8.24. Calibration

**Name:** CALIB
**Offset:** 0x48
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | BIASREFBUF[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | BIASR2R[2:0] | | | | BIASCOMP[2:0] | | |
| Access | | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 10:8 – BIASREFBUF[2:0]** Bias Reference Buffer Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.
The value must be copied only, and must not be changed.

**Bits 6:4 – BIASR2R[2:0]** Bias R2R ampli Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.
The value must be copied only, and must not be changed

**Bits 2:0 – BIASCOMP[2:0]** Bias Comparator Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy. Refer to *NVM Software Calibration Area Mapping* for further details.
The value must be copied only, and must not be changed


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 46. AC – Analog Comparators

## 46.1. Overview

The Analog Comparator (AC) supports two individual comparators. Each comparator (COMP) compares the voltage levels on two inputs, and provides a digital output based on this comparison. Each comparator may be configured to generate interrupt requests and/or peripheral events upon several different combinations of input change.

Hysteresis can be adjusted to achieve the optimal operation for each application.

The input selection includes four shared analog port pins and several internal signals. Each Comparator Output state can also be output on a pin for use by external devices.

The comparators are grouped in pairs on each port. The AC peripheral implements one pair of comparators. These are called Comparator 0 (COMP0) and Comparator 1 (COMP1) . They have identical behaviors, but separate Control registers. The pair can be set in Window mode to compare a signal to a voltage range instead of a single voltage level.

## 46.2. Features

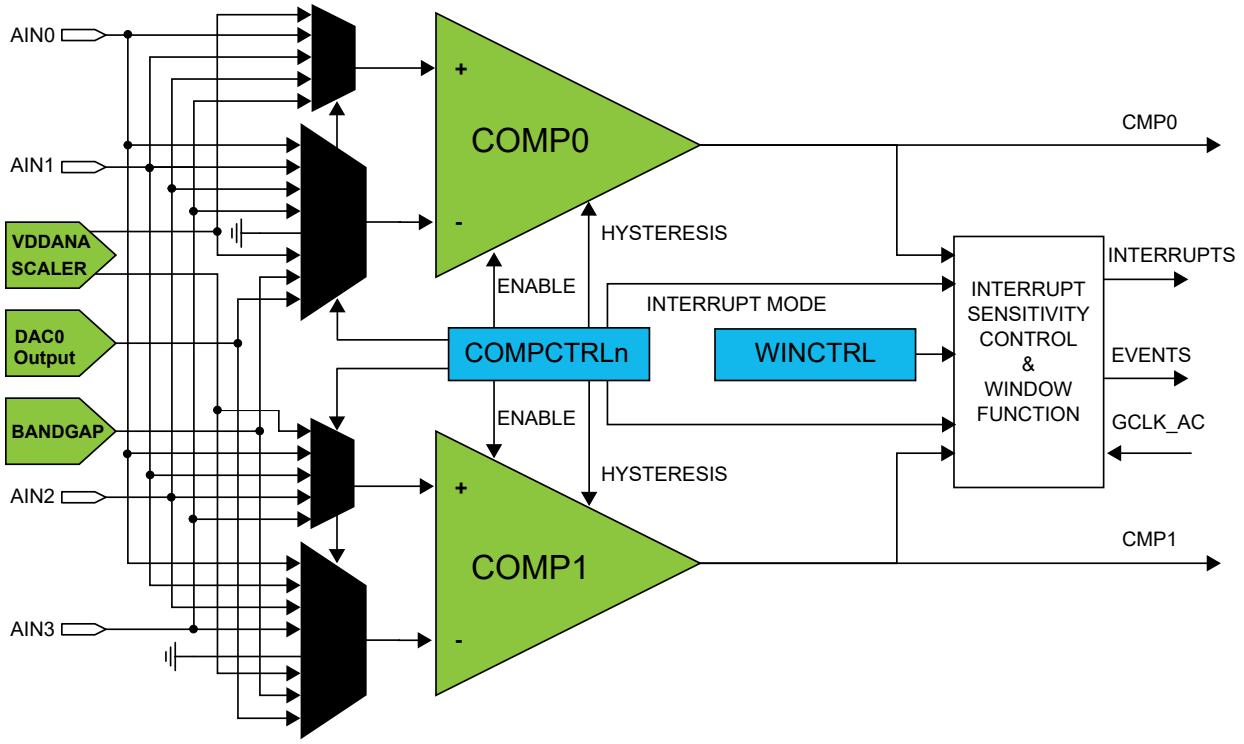
The following are key features of the AC:

- Up to Two individual comparators
- Selectable hysteresis: 3-level On or Off
- Hysteresis: On or Off
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Flexible input selection:
  - Four pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Bandgap reference voltage
  - 64-level programmable VDDANA scaler per comparator
  - DAC0
- Interrupt generation on:
  - Rising or falling edge
  - Toggle
  - End of comparison
- Window function interrupt generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
  - Signal outside window
- Event generation on:
  - Comparator output
  - Window function inside/outside window
- Optional digital filter on comparator output


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 46.3. Block Diagram

**Figure 46-1.** Analog Comparator Block Diagram




## 46.4. Signal Description

| Signal | Description | Type |
|---|---|---|
| AIN[3..0] | Analog input | Comparator inputs |
| CMP[1..0] | Digital output | Comparator outputs |

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

**Related Links**
I/O Multiplexing and Considerations

## 46.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 46.5.1. I/O Lines

Using the AC's I/O lines requires the I/O pins to be configured. Refer to *PORT - I/O Pin Controller* for details.

**Table 46-1.** I/O Lines

| Instance | Signal | I/O Line | Peripheral Function |
|---|---|---|---|
| AC0 | AIN0 | PAxx | A |
| AC0 | AIN1 | PAxx | A |
| AC0 | AIN2 | PAxx | A |
| AC0 | AIN3 | PAxx | A |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 46-1.** I/O Lines (continued)

| Instance | Signal | I/O Line | Peripheral Function |
|---|---|---|---|
| AC0 | CMP0 | PAxx | A |
| AC0 | CMP1 | PAxx | A |

**Related Links**
PORT - I/O Pin Controller

### 46.5.2. Power Management

The AC will continue to operate in any Sleep mode where the selected source clock is running. The AC's interrupts can be used to wake up the device from Sleep modes. Events connected to the Event System can trigger other operations in the system without exiting Sleep modes.

### 46.5.3. Clocks

The AC bus clock (CLK_AC_APB) can be enabled and disabled in the Main Clock module, MCLK (see *MCLK - Main Clock*, and the default state of CLK_AC_APB can be found in *Peripheral Clock Masking*.

A generic clock (GCLK_AC) is required to clock the AC. This clock must be configured and enabled in the generic clock controller before using the AC. Refer to the Generic Clock Controller chapter for details.

This generic clock is asynchronous to the bus clock (CLK_AC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**
Peripheral Clock Masking
MCLK – Main Clock

### 46.5.4. DMA

Not applicable.

### 46.5.5. Interrupts

The interrupt request lines are connected to the interrupt controller. Using the AC interrupts requires the interrupt controller to be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller

### 46.5.6. Events

The events are connected to the Event System. Refer to *EVSYS – Event System* for details on how to configure the Event System.

**Related Links**
Event System (EVSYS)

### 46.5.7. Debug Operation

When the CPU is halted in Debug mode, the AC will halt normal operation after any on-going comparison is completed. The AC can be forced to continue normal operation during debugging. Refer to DBGCTRL for additional information. If the AC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- The Control B register (CTRLB)
- The Interrupt Flag register (INTFLAG)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 46.5.9. Analog Connections

Each comparator has up to four I/O pins that can be used as analog inputs. Each pair of comparators shares the same four pins. These pins must be configured for analog operation before using them as comparator inputs.

Any internal reference source, such as a bandgap voltage reference, or DAC must be configured and enabled prior to its use as a comparator input.

### 46.5.10. Calibration

The BIAS calibration value from the production test must be loaded from the NVM Software Calibration Area into the AC Calibration register (CALIB) by software to achieve specified accuracy.

## 46.6. Functional Description

### 46.6.1. Principle of Operation

Each comparator has one positive input and one negative input. Each positive input may be chosen from a selection of analog input pins. Each negative input may be chosen from a selection of both analog input pins and internal inputs, such as a bandgap voltage reference.

The digital output from the comparator is '1' when the difference between the positive and the negative input voltage is positive, and '0' otherwise.

The individual comparators can be used independently (Normal mode) or paired to form a window comparison (Window mode).

### 46.6.2. Basic Operation

#### 46.6.2.1. Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected: The Event Control register (EVCTRL).

Enable-protection is denoted by the "Enable-Protected" property in each individual register description.

#### 46.6.2.2. Enabling, Disabling and Resetting

The AC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The AC is disabled writing a '0' to CTRLA.ENABLE.

The AC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the AC will be reset to their initial state, and the AC will be disabled. Refer to *CTRLA* for details.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 46.6.2.3. Comparator Configuration

Each individual comparator must be configured by its respective Comparator Control register (COMPCTRLx) before that comparator is enabled. These settings cannot be changed while the comparator is enabled.

- Select the desired measurement mode with COMPCTRLx.SINGLE. See Starting a Comparison for more details.
- Select the desired hysteresis with COMPCTRLx.HYSTEN and COMPCTRLx.HYST. See Input Hysteresis for more details.
- Write COMPCTRLx.SPEED.
- Select the interrupt source with COMPCTRLx.INTSEL.
- Select the positive and negative input sources with the COMPCTRLx.MUXPOS and COMPCTRLx.MUXNEG bits. See Selecting Comparator Inputs for more details.
- Select the filtering option with COMPCTRLx.FLEN.
- Select standby operation with the Run in Standby bit (COMPCTRLx.RUNSTDBY).

The individual comparators are enabled by writing a '1' to the Enable bit in the Comparator x Control registers (COMPCTRLx.ENABLE). The individual comparators are disabled by writing a '0' to COMPCTRLx.ENABLE. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLx.ENABLE bits.

#### 46.6.2.4. Starting a Comparison

Each comparator channel can be in one of two different measurement modes, determined by the Single bit in the Comparator x Control register (COMPCTRLx.SINGLE):

- Continuous measurement
- Single-shot

After being enabled, a start-up delay is required before the result of the comparison is ready. This start-up time is measured automatically to account for environmental changes, such as temperature or voltage supply level, and is specified in the *Electrical Characteristics* chapters. During the start-up time, the COMP output is not available.

The comparator can be configured to generate interrupts when the output toggles, when the output changes from '0' to '1' (rising edge), when the output changes from '1' to '0' (falling edge) or at the end of the comparison. An end-of-comparison interrupt can be used with the Single-Shot mode to chain further events in the system, regardless of the state of the comparator outputs. The Interrupt mode is set by the Interrupt Selection bit group in the Comparator Control register (COMPCTRLx.INTSEL). Events are generated using the comparator output state, regardless of whether the interrupt is enabled or not.

##### 46.6.2.4.1. Continuous Measurement

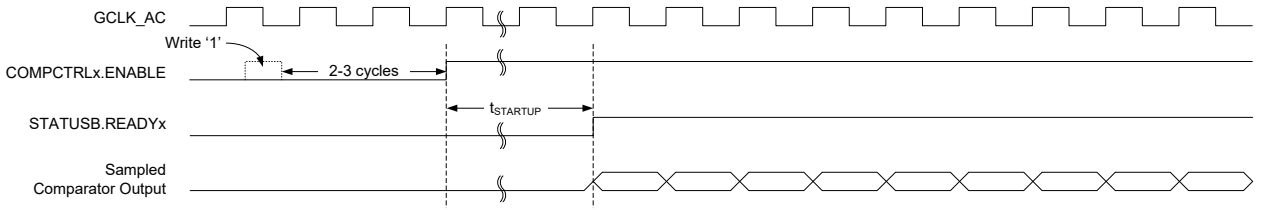
Continuous measurement is selected by writing COMPCTRLx.SINGLE to zero. In continuous mode, the comparator is continuously enabled and performing comparisons. This ensures that the result of the latest comparison is always available in the Current State bit in the Status A register (STATUSA.STATEx).

After the start-up time has passed, a comparison is done and STATUSA is updated. The Comparator x Ready bit in the Status B register (STATUSB.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPCTRLx.ENABLE bit is written to zero. The start-up time applies only to the first comparison.

In continuous operation, edge detection of the comparator output for interrupts is done by comparing the current and previous sample. The sampling rate is the GCLK_AC frequency. An example of continuous measurement is shown in the Figure 46-2.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 46-2.** Continuous Measurement Example




For low-power operation, comparisons can be performed during sleep modes without a clock. The comparator is enabled continuously, and changes of the comparator state are detected asynchronously. When a toggle occurs, the Power Manager will start GCLK_AC to register the appropriate peripheral events and interrupts. The GCLK_AC clock is then disabled again automatically, unless configured to wake up the system from sleep.

#### 46.6.2.4.2. Single-Shot

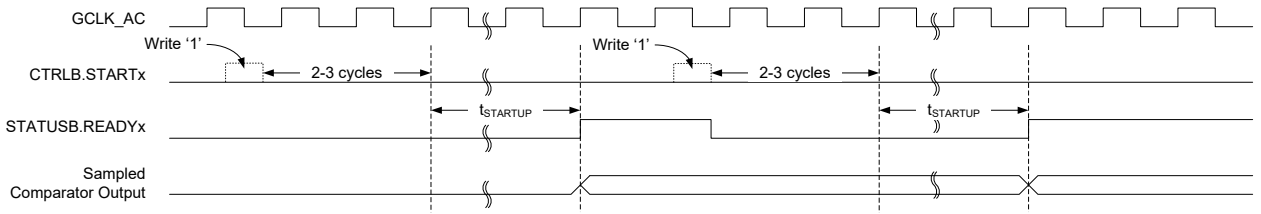
Single-shot operation is selected by writing COMPCTRLx.SINGLE to '1'. During single-shot operation, the comparator is normally idle. The user starts a single comparison by writing '1' to the respective Start Comparison bit in the write-only Control B register (CTRLB.STARTx). The comparator is enabled, and after the start-up time has passed, a single comparison is done and STATUSA is updated. Appropriate peripheral events and interrupts are also generated. No new comparisons will be performed.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status B register (STATUSB.READYx). STATUSB.READYx is set automatically by hardware when the single comparison has completed.

A single-shot measurement can also be triggered by the Event System. Setting the Comparator x Event Input bit in the Event Control Register (EVCTRL.COMPEIx) enables triggering on incoming peripheral events. Each comparator can be triggered independently by separate events. Event-triggered operation is similar to user-triggered operation; the difference is that a peripheral event from another hardware module causes the hardware to automatically start the comparison and will not clear STATUSB.READYx.

To detect an edge of the comparator output in single-shot operation for the purpose of interrupts, the result of the current measurement is compared with the result of the previous measurement (one sampling period earlier). An example of single-shot operation is shown in Figure 46-3.

**Figure 46-3.** Single-Shot Example




For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the startup time has passed, a comparison is done and appropriate peripheral events and interrupts are also generated. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake up the system from sleep.

### 46.6.3. Selecting Comparator Inputs

Each comparator has one positive and one negative input. The positive input is one of the external input pins (AINx). The negative input can be fed either from an external input pin (AINx) or from one


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

of the several internal reference voltage sources common to all comparators. The user selects the input source as follows:

- The positive input is selected by the Positive Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXPOS)
- The negative input is selected by the Negative Input MUX Select bit group in the Comparator Control register (COMPCTRLx.MUXNEG)

In the case of using an external I/O pin, the selected pin must be configured for analog use in the PORT Controller by disabling the digital input and output. The switching of the analog input multiplexers is controlled to minimize crosstalk between the channels. The input selection must be changed only while the individual comparator is disabled.

**Note:** For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

### 46.6.4. Window Operation

Each comparator pair can be configured to work together in Window mode. In this mode, a voltage range is defined, and the comparators give information about whether an input signal is within this range or not. Window mode is enabled by the Window Enable x bit in the Window Control register (WINCTRL.WENx). Both comparators in a pair must have the same measurement mode setting in their respective Comparator Control Registers (COMPCTRLx.SINGLE).

**Notes:**

1. Both comparators must be enabled (COMPCTRLx.ENABLE = 1) before the Window mode is enabled (WINCTRL.WEN0 = 1).
2. Window mode must be first disabled (WINCTRL.WEN0 = 0) before both comparators are disabled (COMPCTRLx.ENABLE = 0).
3. The ready bits for both comparators must be checked (STATUSB.READYx = 1) before reading the measurement results.

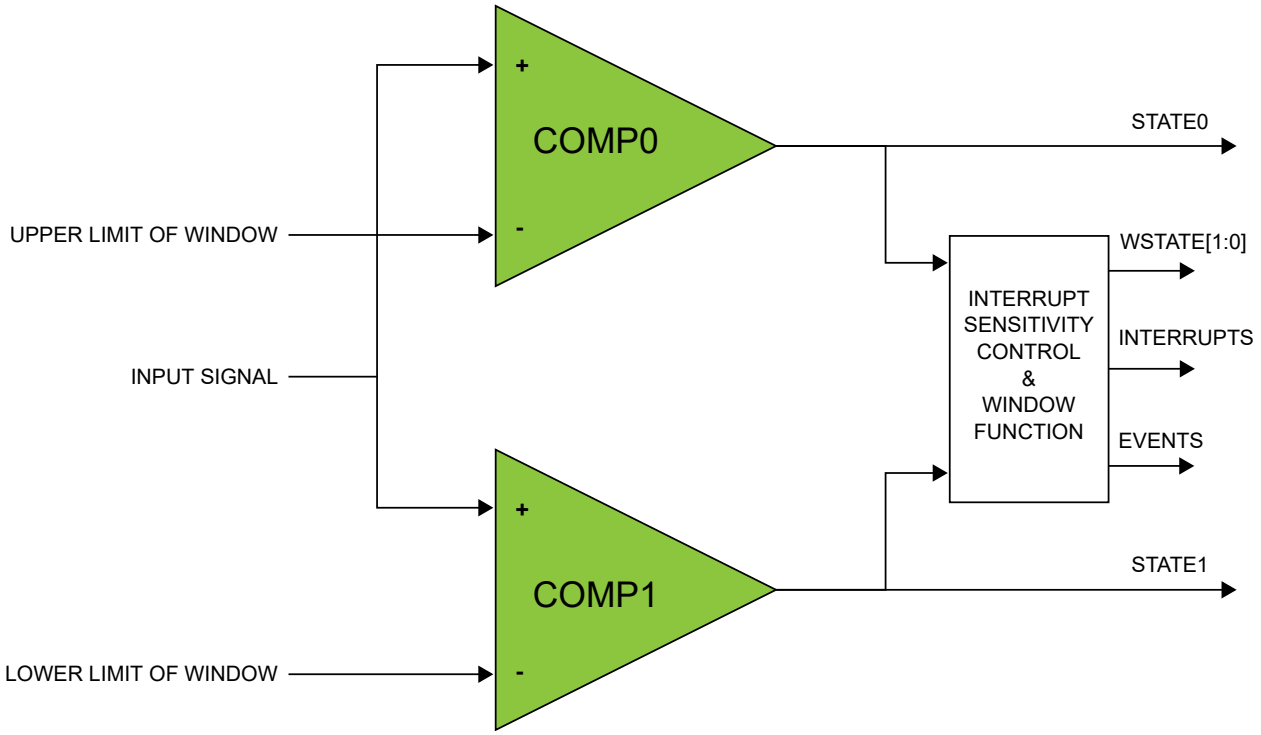
To physically configure the pair of comparators for Window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In the figure below, COMP0 defines the upper limit and COMP1 defines the lower limit of the window, as shown but the window will also work in the opposite configuration with COMP0 lower and COMP1 higher. The current state of the window function is available in the Window x State bit group of the Status register (STATUS.WSTATEx).

Window mode can be configured to generate interrupts when the input voltage changes to below the window, when the input voltage changes to above the window, when the input voltage changes into the window or when the input voltage changes outside the window. The interrupt selections are set by the Window Interrupt Selection bit field in the Window Control register (WINCTRL.WINTSEL). Events are generated using the inside/outside state of the window, regardless of whether the interrupt is enabled or not. Note that the individual comparator outputs, interrupts and events continue to function normally during Window mode.

When the comparators are configured for Window mode and Single-shot mode, measurements are performed simultaneously on both comparators. Writing '1' to either Start Comparison bit in the Control B register (CTRLB.STARTx) will start a measurement. Likewise either peripheral event can start a measurement.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 46-4.** Comparators in Window Mode

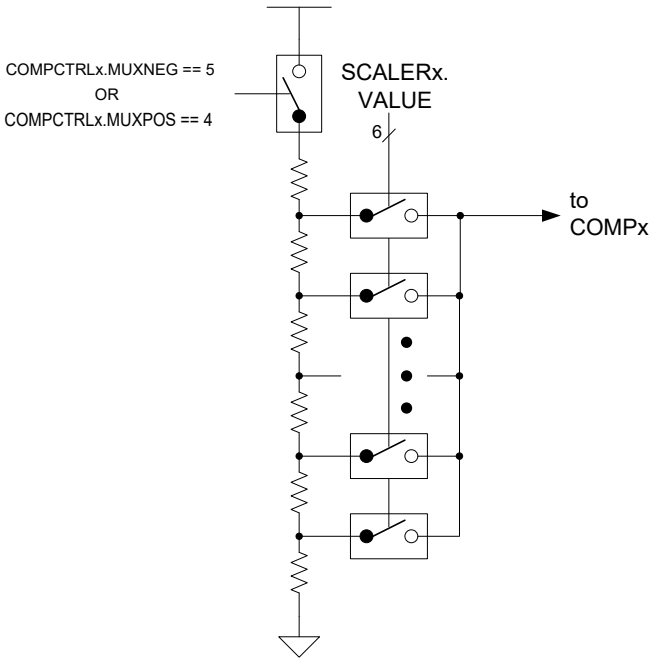



### 46.6.5. VDDANA Scaler

The VDDANA scaler generates a reference voltage that is a fraction of the device's supply voltage, with 64 levels. One independent voltage channel is dedicated for each comparator. The scaler of a comparator is enabled when the Negative Input Mux bit field or the Positive Input Mux in the respective Comparator Control register (COMPCTRLx) is set to VSCALE as an input and the comparator is enabled. The voltage of each channel is selected by the Value bit field in the SCALERx registers (SCALERx.VALUE).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 46-5.** VDDANA Scaler




### 46.6.6. Input Hysteresis

Application software can selectively enable/disable hysteresis for the comparison. Applying hysteresis will help prevent constant toggling of the output, which can be caused by noise when the input signals are close to each other.

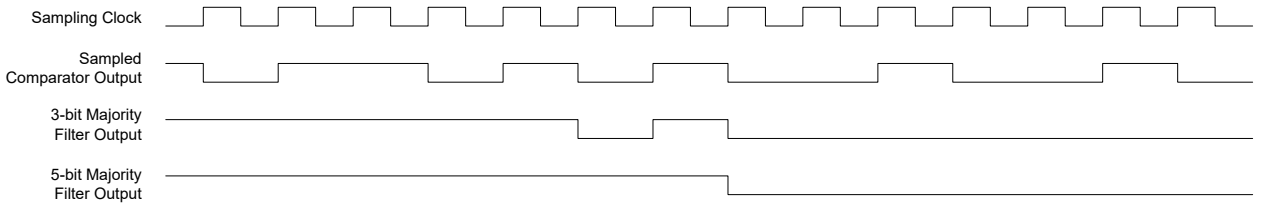
Hysteresis is enabled for each comparator individually by the Hysteresis Enable bit in the Comparator x Control register (COMPCTRLx.HYSTEN). Furthermore, when enabled, the level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLx.HYST). Hysteresis is available only in Continuous mode (COMPCTRLx.SINGLE=0).

### 46.6.7. Filtering

The output of the comparators can be filtered digitally to reduce noise. The filtering is determined by the Filter Length bits in the Comparator Control x register (COMPCTRLx.FLEN), and is independent for each comparator. Filtering is selectable from none, 3-bit majority (N=3) or 5-bit majority (N=5) functions. Any change in the comparator output is considered valid only if N/2+1 out of the last N samples agree. The filter sampling rate is the GCLK_AC frequency.

Note that filtering creates an additional delay of N-1 sampling cycles from when a comparison is started until the comparator output is validated. For Continuous mode, the first valid output will occur when the required number of filter samples is taken. Subsequent outputs will be generated every cycle based on the current sample plus the previous N-1 samples, as shown in Figure 46-6. For Single-shot mode, the comparison completes after the Nth filter sample, as shown in Figure 46-7.

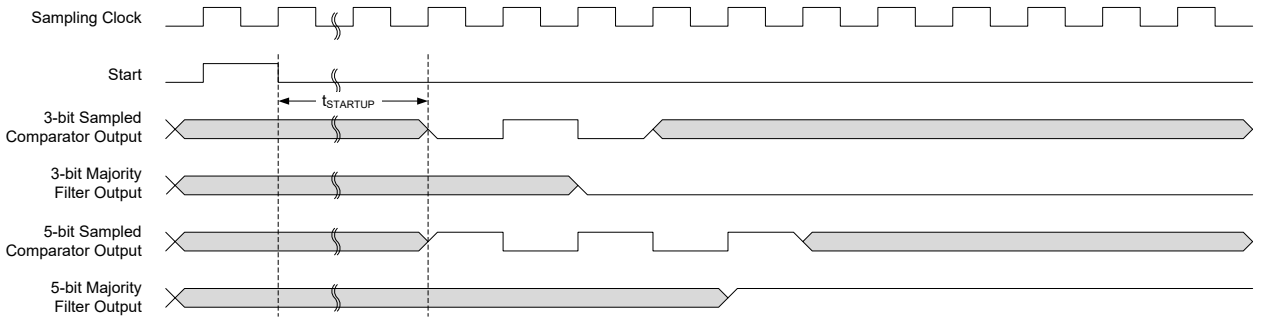
**Figure 46-6.** Continuous Mode Filtering





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 46-7.** Single-Shot Filtering




During Sleep modes, filtering is supported only for single-shot measurements. Filtering must be disabled if continuous measurements will be done during Sleep modes, or the resulting interrupt/event may be generated incorrectly.

### 46.6.8. Comparator Output

The output of each comparator can be routed to an I/O pin by setting the Output bit group in the Comparator Control x register (COMPCTRLx.OUT). This allows the comparator to be used by external circuitry. Either the raw, non-synchronized output of the comparator or the CLK_AC-synchronized version, including filtering, can be used as the I/O signal source. The output appears on the corresponding CMP[x] pin.

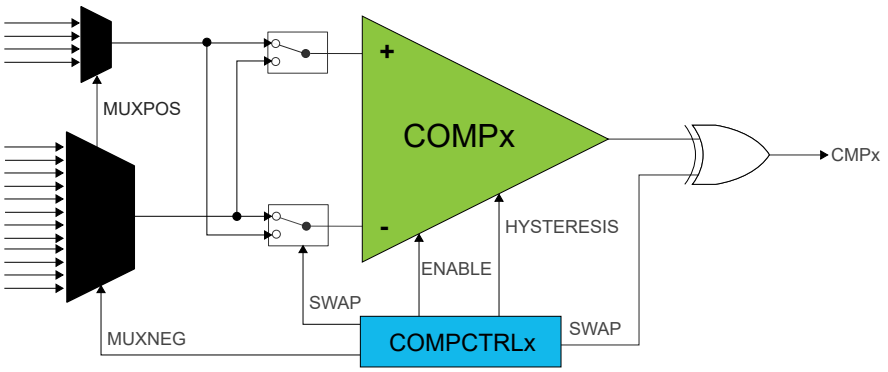
### 46.6.9. Offset Compensation

The Swap bit in the Comparator Control registers (COMPCTRLx.SWAP) controls switching of the input signals to a comparator's positive and negative terminals. When the comparator terminals are swapped, the output signal from the comparator is also inverted, as shown in the following figure.

**Note:** The output signal from the comparator is inverted only if the comparator inputs difference is within offset values.

This allows the user to measure or compensate for the comparator input offset voltage. As part of the input selection, COMPCTRLx.SWAP can be changed only while the comparator is disabled.

**Figure 46-8.** Input Swapping for Offset Compensation




### 46.6.10. DMA Operation

Not applicable.

### 46.6.11. Interrupts

The AC has the following interrupt sources:

- Comparator (COMP0, COMP1): Indicates a change in comparator status.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Window (WIN0): Indicates a change in the window status.

Comparator interrupts are generated based on the conditions selected by the Interrupt Selection bit group in the Comparator Control registers (COMPCTRLx.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSEL[1:0]).

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the AC is Reset. See INFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

**Related Links**

Nested Vector Interrupt Controller

### 46.6.12. Events

The AC can generate the following output events:

- Comparator (COMP0, COMP1): Generated as a copy of the comparator status
- Window (WIN0): Generated as a copy of the window inside/outside status

Writing a one to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables the corresponding output event. Writing a zero to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The AC can take the following action on an input event:

- Start comparison (START0, START1): Start a comparison.

Writing a one to an Event Input bit into the Event Control register (EVCTRL.COMPEIx) enables the corresponding action on input event. Writing a zero to this bit disables the corresponding action on input event. Note that if several events are connected to the AC, the enabled action will be taken on any of the incoming events. Refer to the Event System chapter for details on configuring the event system.

When EVCTRL.COMPEIx is one, the event will start a comparison on COMPx after the start-up time delay. In normal mode, each comparator responds to its corresponding input event independently. For a pair of comparators in window mode, either comparator event will trigger a comparison on both comparators simultaneously.

### 46.6.13. Sleep Mode Operation

The Run in Standby bits in the Comparator x Control registers (COMPCTRLx.RUNSTDBY) control the behavior of the AC during standby sleep mode. Each RUNSTDBY bit controls one comparator. When the bit is zero, the comparator is disabled during sleep, but maintains its current configuration. When the bit is one, the comparator continues to operate during sleep. Note that when RUNSTDBY is zero, the analog blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

For Window Mode operation, both comparators in a pair must have the same RUNSTDBY configuration.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When RUNSTDBY is one, any enabled AC interrupt source can wake up the CPU. The AC can also be used during sleep modes where the clock used by the AC is disabled, provided that the AC is still powered (not in shutdown). In this case, the behavior is slightly different and depends on the measurement mode, as listed in Table 46-2.

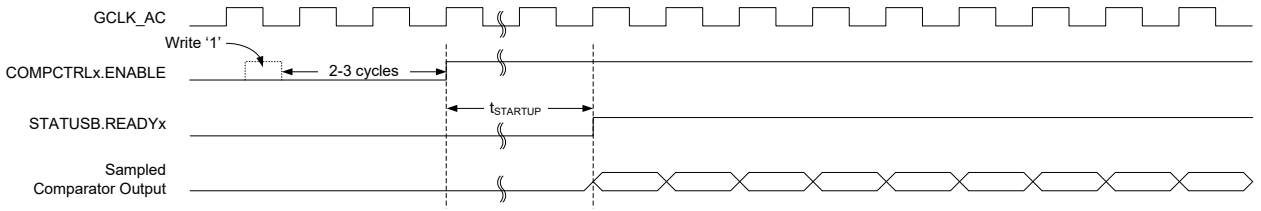
**Table 46-2.** Sleep Mode Operation

| COMPCTRLx.MODE | RUNSTDBY=0 | RUNSTDBY=1 |
|---|---|---|
| 0 (Continuous) | COMPx disabled | GCLK_AC stopped, COMPx enabled |
| 1 (Single-shot) | COMPx disabled | GCLK_AC stopped, COMPx enabled only when triggered by an input event |

### 46.6.13.1. Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and GCLK_AC is disabled during sleep, the comparator will remain continuously enabled and will function asynchronously. The current state of the comparator is asynchronously monitored for changes. If an edge matching the interrupt condition is found, GCLK_AC is started to register the interrupt condition and generate events. If the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device; otherwise GCLK_AC is disabled until the next edge detection. Filtering is not possible with this configuration.

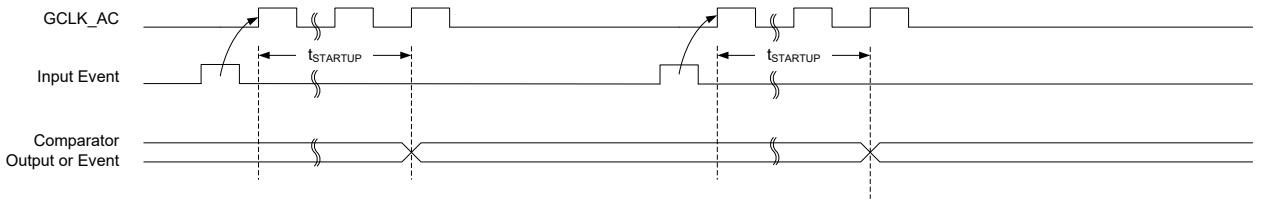
**Figure 46-9.** Continuous Mode SleepWalking




### 46.6.13.2. Single-Shot Measurement during Sleep

For low-power operation, event-triggered measurements can be performed during sleep modes. When the event occurs, the Power Manager will start GCLK_AC. The comparator is enabled, and after the start-up time has passed, a comparison is done, with filtering if desired, and the appropriate peripheral events and interrupts are also generated, as shown in Figure 46-10. The comparator and GCLK_AC are then disabled again automatically, unless configured to wake the system from sleep. Filtering is allowed with this configuration.

**Figure 46-10.** Single-Shot SleepWalking




## 46.6.14. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in Control register (CTRLA.SWRST)
- Enable bit in Control register (CTRLA.ENABLE)
- Enable bit in Comparator Control register (COMPCTRLn.ENABLE)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The following registers are synchronized when written:

- Window Control register (WINCTRL)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

**Related Links**

Register Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

## 46.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | | ENABLE | SWRST |
| 0x01 | CTRLB | 7:0 | | | | | | | START1 | START0 |
| 0x02 | EVCTRL | 7:0 | | | | WINEO0 | | | COMPEO1 | COMPEO0 |
| | | 15:8 | | | INVEI1 | INVEI0 | | | COMPEI1 | COMPEI0 |
| 0x04 | INTENCLR | 7:0 | | | | WIN0 | | | COMP1 | COMP0 |
| 0x05 | INTENSET | 7:0 | | | | WIN0 | | | COMP1 | COMP0 |
| 0x06 | INTFLAG | 7:0 | | | | WIN0 | | | COMP1 | COMP0 |
| 0x07 | STATUSA | 7:0 | | | WSTATE0[1:0] | | | | STATE1 | STATE0 |
| 0x08 | STATUSB | 7:0 | | | | | | | READY1 | READY0 |
| 0x09 | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0A | WINCTRL | 7:0 | | | | | | WINTSEL0[1:0] | | WEN0 |
| 0x0B | Reserved | | | | | | | | | |
| 0x0C | SCALER0 | 7:0 | | | VALUE[5:0] | | | | | |
| 0x0D | SCALER1 | 7:0 | | | VALUE[5:0] | | | | | |
| 0x0E ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | COMPCTRL0 | 7:0 | | RUNSTDBY | | INTSEL[1:0] | | SINGLE | ENABLE | |
| | | 15:8 | SWAP | | MUXPOS[2:0] | | | | MUXNEG[2:0] | |
| | | 23:16 | | | HYST[1:0] | | HYSTEN | | SPEED[1:0] | |
| | | 31:24 | | | OUT[1:0] | | | | FLEN[2:0] | |
| 0x14 | COMPCTRL1 | 7:0 | | RUNSTDBY | | INTSEL[1:0] | | SINGLE | ENABLE | |
| | | 15:8 | SWAP | | MUXPOS[2:0] | | | | MUXNEG[2:0] | |
| | | 23:16 | | | HYST[1:0] | | HYSTEN | | SPEED[1:0] | |
| | | 31:24 | | | OUT[1:0] | | | | FLEN[2:0] | |
| 0x18 ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | SYNCBUSY | 7:0 | | | | COMPCTRL1 | COMPCTRL0 | WINCTRL | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | CALIB | 7:0 | | | | | | | BIAS0[1:0] | |
| | | 15:8 | | | | | | | | |

## 46.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R/W | W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ENABLE** Enable

Due to synchronization, there is delay from updating the register until the peripheral is enabled/disabled. The value written to CTRL.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE is cleared when the peripheral is enabled/disabled.

**Note:** To avoid spurious interrupts from enable/disable cycles, use the SWRST bit to reset the comparator module.

| Value | Description |
|---|---|
| 0 | The AC is disabled. |
| 1 | The AC is enabled. Each comparator must also be enabled individually by the Enable bit in the Comparator Control register (COMPCTRLn.ENABLE). |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the AC to their initial state, and the AC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.2. Control B

**Name:** CTRLB
**Offset:** 0x01
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | START1 | START0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 0, 1 – STARTx** Comparator x Start Comparison
Writing a '0' to this field has no effect.
Writing a '1' to STARTx starts a single-shot comparison on COMPx if both the Single-Shot and Enable bits in the Comparator x Control Register are '1' (COMPCTRLx.SINGLE and COMPCTRLx.ENABLE). If comparator x is not implemented, or if it is not enabled in single-shot mode, Writing a '1' has no effect.
This bit always reads as zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.3. Event Control

**Name:** EVCTRL
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | INVEI1 | INVEI0 | | | COMPEI1 | COMPEI0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | WINEO0 | | | COMPEO1 | COMPEO0 |
| Access | | | | R/W | | | R/W | R/W |
| Reset | | | | 0 | | | 0 | 0 |

**Bits 12, 13 – INVEIx** Inverted Event Input Enable x

| Value | Description |
|---|---|
| 0 | Incoming event is not inverted for comparator x. |
| 1 | Incoming event is inverted for comparator x. |

**Bits 8, 9 – COMPEIx** Comparator x Event Input
Note that several actions can be enabled for incoming events. If several events are connected to the peripheral, the enabled action will be taken for any of the incoming events. There is no way to tell which of the incoming events caused the action.
These bits indicate whether a comparison will start or not on any incoming event.

| Value | Description |
|---|---|
| 0 | Comparison will not start on any incoming event. |
| 1 | Comparison will start on any incoming event. |

**Bit 4 – WINEO0** Window 0 Event Output Enable
These bits indicate whether the window 0 function can generate a peripheral event or not.

| Value | Description |
|---|---|
| 0 | Window 0 Event is disabled. |
| 1 | Window 0 Event is enabled. |

**Bits 0, 1 – COMPEOx** Comparator x Event Output Enable
These bits indicate whether the comparator x output can generate a peripheral event or not.

| Value | Description |
|---|---|
| 0 | COMPx event generation is disabled. |
| 1 | COMPx event generation is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | WIN0 | | | COMP1 | COMP0 |
| Access | | | | R/W | | | R/W | R/W |
| Reset | | | | 0 | | | 0 | 0 |

**Bit 4 – WIN0** Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Window 0 interrupt.

| Value | Description |
|---|---|
| 0 | The Window 0 interrupt is disabled. |
| 1 | The Window 0 interrupt is enabled. |

**Bits 0, 1 – COMPx** Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit disables the Comparator x interrupt.

| Value | Description |
|---|---|
| 0 | The Comparator x interrupt is disabled. |
| 1 | The Comparator x interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | WIN0 | | | COMP1 | COMP0 |
| Access | | | | R/W | | | R/W | R/W |
| Reset | | | | 0 | | | 0 | 0 |

**Bit 4 – WIN0** Window 0 Interrupt Enable
Reading this bit returns the state of the Window 0 interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit enables the Window 0 interrupt.

| Value | Description |
|---|---|
| 0 | The Window 0 interrupt is disabled. |
| 1 | The Window 0 interrupt is enabled. |

**Bits 0, 1 – COMPx** Comparator x Interrupt Enable
Reading this bit returns the state of the Comparator x interrupt enable.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Ready interrupt bit and enable the Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Comparator x interrupt is disabled. |
| 1 | The Comparator x interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x06
**Reset:** 0x00
**Property:** –

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | WIN0 | | | COMP1 | COMP0 |
| Access | | | | R/W | | | R/W | R/W |
| Reset | | | | 0 | | | 0 | 0 |

**Bit 4 – WIN0** Window 0
This flag is set according to the Window 0 Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and will generate an interrupt if INTENCLR/SET.WINx is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Window 0 interrupt flag.

**Bits 0, 1 – COMPx** Comparator x
Reading this bit returns the status of the Comparator x interrupt flag. If comparator x is not implemented, COMPx always reads as zero.
This flag is set according to the Interrupt Selection bit group in the Comparator x Control register (COMPCTRLx.INTSEL) and will generate an interrupt if INTENCLR/SET.COMPx is also one.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Comparator x interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.7. Status A

**Name:** STATUSA
**Offset:** 0x07
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | WSTATE0[1:0] | | | | STATE1 | STATE0 |
| Access | | | R | R | | | R | R |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bits 5:4 – WSTATE0[1:0]** Window 0 Current State
These bits show the current state of the signal if the window 0 mode is enabled.
These values may change in during startup and measurement cycles. When polling for sample completion use both STATUSB.READYx bits to signal completion.

| Value | Name | Description |
|---|---|---|
| 0x0 | ABOVE | Signal is above window |
| 0x1 | INSIDE | Signal is inside window |
| 0x2 | BELOW | Signal is below window |
| 0x3 | - | Reserved |

**Bits 0, 1 – STATEx** Comparator x Current State
This bit shows the current state of the output signal from COMPx. STATEx is valid only when the STATUSB.READYx bit is one.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.8. Status B

**Name:** STATUSB
**Offset:** 0x08
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | READY1 | READY0 |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bits 0, 1 – READYx** Comparator x Ready
This bit is cleared when the comparator x output is not ready.
This bit is set when the comparator x output is ready.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.9. Debug Control

**Name:** DBGCTRL
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
This bit is not reset by a software reset.
This bits controls the functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The AC is halted when the CPU is halted by an external debugger. Any on-going comparison will complete. |
| 1 | The AC continues normal operation when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 46.8.10. Window Control

**Name:** WINCTRL
**Offset:** 0x0A
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WINTSEL0[1:0] | | WEN0 |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 2:1 – WINTSEL0[1:0]** Window 0 Interrupt Selection
These bits configure the interrupt mode for the comparator window 0 mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | ABOVE | Interrupt on signal above window |
| 0x1 | INSIDE | Interrupt on signal inside window |
| 0x2 | BELOW | Interrupt on signal below window |
| 0x3 | OUTSIDE | Interrupt on signal outside window |

**Bit 0 – WEN0** Window 0 Mode Enable

| Value | Description |
|---|---|
| 0 | Window mode is disabled for comparators 0 and 1. |
| 1 | Window mode is enabled for comparators 0 and 1. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.11. Scaler n

**Name:** SCALER
**Offset:** 0x0C + n*0x01 [n=0..1]
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | VALUE[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:0 – VALUE[5:0]** Scaler Value
These bits define the scaling factor for channel n of the VDDANA voltage scaler. The output voltage, $V_{SCALE}$, is:

$$V_{SCALE} = \frac{V_{DD} \cdot (VALUE+1)}{64}$$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.12. Comparator Control n

**Name:** COMPCTRLn
**Offset:** 0x10 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | OUT[1:0] | | | FLEN[2:0] | | |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 0 | 0 | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | HYST[1:0] | | HYSTEN | | SPEED[1:0] | |
| Access | | | R/W | R/W | R/W | | R/W | R/W |
| Reset | | | 0 | 0 | 0 | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SWAP | MUXPOS[2:0] | | | | MUXNEG[2:0] | | |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | RUNSTDBY | | INTSEL[1:0] | | SINGLE | ENABLE | |
| Access | | R/W | | R/W | R/W | R/W | R/W | |
| Reset | | 0 | | 0 | 0 | 0 | 0 | |

**Bits 29:28 – OUT[1:0]** Output
These bits configure the output selection for comparator n. COMPCTRLn.OUT can be written only while COMPCTRLn.ENABLE is zero.
**Note:** For internal use of the comparison results by the CCL, this bit must be 0x1 or 0x2.

These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | The output of COMPn is not routed to the COMPn I/O port |
| 0x1 | ASYNC | The asynchronous output of COMPn is routed to the COMPn I/O port |
| 0x2 | SYNC | The synchronous output (including filtering) of COMPn is routed to the COMPn I/O port |
| 0x3 | N/A | Reserved |

**Bits 26:24 – FLEN[2:0]** Filter Length
These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | No filtering |
| 0x1 | MAJ3 | 3-bit majority function (2 of 3) |
| 0x2 | MAJ5 | 5-bit majority function (3 of 5) |
| 0x3-0x7 | N/A | Reserved |

**Bits 21:20 – HYST[1:0]** Hysteresis Level
These bits indicate the hysteresis level of comparator n when hysteresis is enabled (COMPCTRLn.HYSTEN = 1). Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE = 0). COMPCTRLn.HYST can be written only while COMPCTRLn.ENABLE is zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | HYST25 | 25mV |
| 0x1 | HYST50 | 50mV |
| 0x2 | HYST75 | 75mV |
| 0x3 | HYST100 | 100mV |

**Bit 19 – HYSTEN** Hysteresis Enable

This bit indicates the hysteresis mode of comparator n. Hysteresis is available only for continuous mode (COMPCTRLn.SINGLE = 0).
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Hysteresis is disabled. |
| 1 | Hysteresis is enabled. |

**Bits 17:16 – SPEED[1:0]** Speed Selection

This bitfield must be written to 0x3 for each comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x3 | HIGH | High speed |
| Other | - | Reserved |

**Bit 15 – SWAP** Swap Inputs and Invert

This bit swaps the positive and negative inputs to COMPn and inverts the output. This function can be used for offset cancellation. COMPCTRLn.SWAP can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

| Value | Description |
|---|---|
| 0 | The output of MUXPOS connects to the positive input, and the output of MUXNEG connects to the negative input. |
| 1 | The output of MUXNEG connects to the positive input, and the output of MUXPOS connects to the negative input. |

**Bits 14:12 – MUXPOS[2:0]** Positive Input Mux Selection

These bits select which input will be connected to the positive input of comparator n. COMPCTRLn.MUXPOS can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | PIN0 | I/O pin 0 |
| 0x1 | PIN1 | I/O pin 1 |
| 0x2 | PIN2 | I/O pin 2 |
| 0x3 | PIN3 | I/O pin 3 |
| 0x4 | VSCALE | VDDANA scaler |
| 0x5-0x7 | - | Reserved |

**Bits 10:8 – MUXNEG[2:0]** Negative Input Mux Selection

These bits select which input will be connected to the negative input of comparator n. COMPCTRLn.MUXNEG can only be written while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | PIN0 | I/O pin 0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x1 | PIN1 | I/O pin 1 |
| 0x2 | PIN2 | I/O pin 2 |
| 0x3 | PIN3 | I/O pin 3 |
| 0x4 | GND | Ground |
| 0x5 | VSCALE | VDDANA scaler |
| 0x6 | BANDGAP | Internal bandgap voltage |
| 0x7 | DAC0 | DAC output |

**Bit 6 – RUNSTDBY** Run in Standby

This bit controls the behavior of the comparator during standby sleep mode.
This bit is not synchronized

| Value | Description |
|---|---|
| 0 | The comparator is disabled during sleep. |
| 1 | The comparator continues to operate during sleep. |

**Bits 4:3 – INTSEL[1:0]** Interrupt Selection

These bits select the condition for comparator n to generate an interrupt or event.
COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | TOGGLE | Interrupt on comparator output toggle |
| 0x1 | RISING | Interrupt on comparator output rising |
| 0x2 | FALLING | Interrupt on comparator output falling |
| 0x3 | EOC | Interrupt on end of comparison (single-shot mode only) |

**Bit 2 – SINGLE** Single-Shot Mode

This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.
These bits are not synchronized.

| Value | Description |
|---|---|
| 0 | Comparator n operates in continuous measurement mode. |
| 1 | Comparator n operates in single-shot mode. |

**Bit 1 – ENABLE** Enable

Writing a zero to this bit disables comparator n.
Writing a one to this bit enables comparator n.
Due to synchronization, there is delay from updating the register until the comparator is enabled/disabled. The value written to COMPCTRLn.ENABLE will read back immediately after being written. SYNCBUSY.COMPCTRLn is set. SYNCBUSY.COMPCTRLn is cleared when the peripheral is enabled/disabled.
Writing a one to COMPCTRLn.ENABLE will prevent further changes to the other bits in COMPCTRLn. These bits remain protected until COMPCTRLn.ENABLE is written to zero and the write is synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.13. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | COMPCTRL1 | COMPCTRL0 | WINCTRL | ENABLE | SWRST |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bits 3, 4 – COMPCTRLx** COMPCTRLx Synchronization Busy
This bit is cleared when the synchronization of the COMPCTRLx register between the clock domains is complete.
This bit is set when the synchronization of the COMPCTRLx register between clock domains is started.

**Bit 2 – WINCTRL** WINCTRL Synchronization Busy
This bit is cleared when the synchronization of the WINCTRL register between the clock domains is complete.
This bit is set when the synchronization of the WINCTRL register between clock domains is started.

**Bit 1 – ENABLE** Enable Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.ENABLE bit between clock domains is started.

**Bit 0 – SWRST** Software Reset Synchronization Busy
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.SWRST bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 46.8.14. Calibration Register

**Name:** CALIB
**Offset:** 0x24
**Reset:** 0x0101
**Property:** Enable-Protect, PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BIAS0[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 1 |

**Bits 1:0 – BIAS0[1:0]** COMP0/1 Bias Scaling
This value from production test must be loaded from the NVM software calibration row into the CALIB register by software to achieve the specified accuracy.The value must be copied only, and must not be changed


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 47. DAC – Digital-to-Analog Converter

## 47.1. Overview

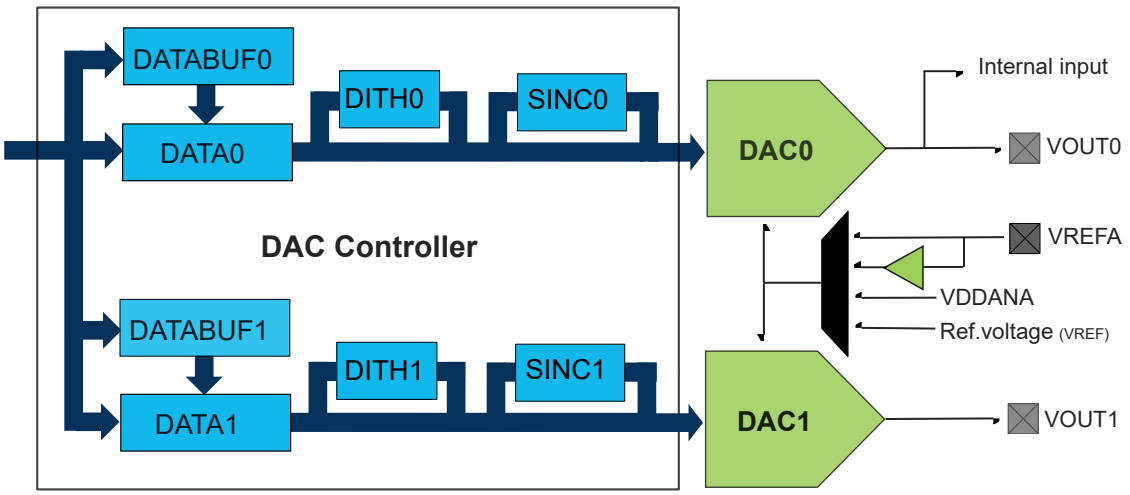
The Digital-to-Analog Converter (DAC) converts a digital value to a voltage. The DAC Controller controls two DACs, which can operate either as two independent DACs or as a single DAC in differential mode. Each DAC is 12-bit resolution and is capable of converting up to 1,000,000 samples per second (MSPS).

## 47.2. Features

- Two independent DACs or single DAC in differential mode
- DAC with 12-bit resolution
- Integrated or Standalone filters with 2x, 4x, 8x, 16x, or 32x oversampling rate (OSR)
- Up to 1MSPS conversion rate
- Hardware support for 16-bit using dithering
- Multiple trigger sources
- High-drive capabilities
- DAC0 used as internal input
- DMA support

## 47.3. Block Diagram

**Figure 47-1.** DAC Controller Block Diagram




## 47.4. Signal Description

| Signal | Description | Type |
|---|---|---|
| VOUT0 | DAC0 output | Analog output |
| VOUT1 | DAC1 output | Analog output |
| VREFA | External reference | Analog input |

One signal can be mapped on several pins.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Important:** Only one analog output function can be used on the same pin.

**Related Links**
I/O Multiplexing and Considerations

## 47.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 47.5.1. I/O Lines

Using the DAC Controller's I/O lines requires the I/O pins to be configured in the PORT - I/O Pin Controller.

**Table 47-1.** I/O Lines

| Instance | Signal | Peripheral Function |
|---|---|---|
| DAC | VOUT0 | A |
| DAC | VOUT1 | A |
| DAC | VREFA | A |

### 47.5.2. Power Management

The DAC Controller will continue to operate in any sleep mode where the selected source clock is running.

The DAC Controller interrupts can be used to wake up the device from sleep modes.

Events connected to the event system can trigger other operations in the system without exiting sleep modes.

**Related Links**
PM – Power Manager

### 47.5.3. Clocks

The DAC bus clock (CLK_DAC_APB) can be enabled and disabled in the Main Clock module, and the default state of CLK_DAC_APB can be found in *Peripheral Clock Masking*.

A generic clock (GCLK_DAC) is required to clock the DAC Controller. This clock must be configured and enabled in the generic clock controller before using the DAC Controller.

This generic clock is asynchronous to the bus clock (CLK_DAC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**
Peripheral Clock Masking
GCLK - Generic Clock Controller

### 47.5.4. DMA

The DMA request line is connected to the DMA Controller (DMAC). Using the DAC Controller DMA requests requires to configure the DMAC first.

**Related Links**
DMAC – Direct Memory Access Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using the DAC Controller interrupts requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 47.5.6. Events

The events are connected to the Event System.

**Related Links**

Event System (EVSYS)

### 47.5.7. Debug Operation

When the CPU is halted in debug mode the DAC will halt normal operation. Any on-going conversions will be completed. The DAC can be forced to continue normal operation during debugging. If the DAC is configured in a way that requires it to be periodically serviced by the CPU through interrupts or similar, improper operation or data loss may result during debugging.

**Related Links**

DBGCTRL

Debug Control

### 47.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag Status and Clear (INTFLAG) register
- Data Buffer (DATABUFx) registers

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**

PAC - Peripheral Access Controller

### 47.5.9. Analog Connections

The DAC has up to two analog output pins (VOUT0, VOUT1) and one analog input pin (VREFA) that must be configured first.

When an internal input is used, it must be enabled before DAC Controller is enabled.

The analog signals of AC, ADC, and DAC can be interconnected.

## 47.6. Functional Description

### 47.6.1. Principle of Operation

Each DAC converts the digital value located in the Data register (DATA0 or DATA1) into an analog voltage on the DAC output (VOUT0 or VOUT1, respectively).

A conversion is started when new data is loaded to the Data register. The resulting voltage is available on the DAC output after the conversion time. A conversion can also be started by input events from Event System.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 47.6.2. Basic Operation

### 47.6.2.1. Initialization

The following registers are enable-protected, meaning they can only be written when the DAC Controller is disabled (CTRLA.ENABLE=0):

- Control B register (CTRLB)
- Event Control register (EVCTRL)
- DAC0 Control (DACCTRL0)
- DAC1 Control (DACCTRL1)

Enable-protection is denoted by the Enable-Protected property in the register description.

### 47.6.2.2. Enabling, Disabling and Resetting

The DAC Controller is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The DAC Controller is disabled by writing a '0' to CTRLA.ENABLE.

The DAC Controller is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the DAC will be reset to their initial state, and the DAC Controller will be disabled. Refer to CTRLA for details.

### 47.6.2.3. DAC Configuration

Each individual DAC is configured by its respective DAC Control register (DACCTRLx)). These settings are applied when DAC Controller is enabled and can be changed only when DAC Controller is disabled.

- Enable the selected DAC by writing a '1' to DACCTRLx.ENABLE.
- Select the data alignment with DACCCTRLx.LEFTADJ. Writing a '1' will left-align the data (DATABUFx/DATAx[31:20]). Writing a '0' to LEFTADJ will right-align the data (DATABUFx/DATAx[11:0]).
- If operation in standby mode is desired for DACx, write a '1' to the Run in Standby bit in the DAC Control register (DACCCTRLx.RUNSTDBY). If RUNSTDBY=1, DACx continues normal operation when system is in standby mode. If RUNSTDBY=0, DACx is halted in standby mode.
- Select dithering mode with DACCCTRLx.DITHER. Writing '1' to DITHER will enable dithering mode, writing a '0' will disable it. Refer to Dithering Mode for details.
- Select the refresh period with the Refresh Period bit field in DACCCTRLx.REFRESH[3:0]. Writing any value greater than '1' to the REFRESH bit field will enable and select the refresh mode. Refer to Conversion Refresh for details.
- Select the output buffer current according to data rate (for low power application) with the Current Control bit field DACCTRLx.CCTRL[1:0]. Refer to Output Buffer Current Control for details.
- Select standalone filter usage by writing to DACCTRLx.FEXT. Writing FEXT=1 selects a standalone filter, FEXT=0 selects the filter integrated to the DAC. See also Interpolation Mode for details.
- Select the filter oversampling ratio by writing to DACCTRLx.OSR[2:0]. writing OSR=0 selects no oversampling; writing any other value will enable interpolation of input data. See also Interpolation Mode for details.

Once the DAC Controller is enabled, DACx requires a startup time before the first conversion can start. The DACx Startup Ready bit in the Status register (STATUS.READYx) indicates that DACx is ready to convert a data when STATUS.READYx=1.

Conversions started while STATUS.READYx=0 shall be discarded.

VOUTx is at tri-state level if DACx is not enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.6.2.4. Digital to Analog Conversion

Each DAC converts a digital value (stored in DATAx register) into an analog voltage. The conversion range is between GND and the selected DAC voltage reference VREF. The default source for VREF is the internal reference voltage VREF. Other voltage reference options are the analog supply voltage (VDDANA) and the external voltage reference (VREFA). The voltage reference is selected by writing to the Reference Selection bits in the Control B register (CTRLB.REFSEL).

The output voltage from the DAC can be calculated using the following formula:

$$V_{OUTx} = \frac{DATAx}{4095} \times VREF$$

A new conversion starts as soon as a new value is loaded into DATAx. DATAx can either be loaded via the APB bus during a CPU write operation, using DMA, or from the DATABUFx register when a STARTx event occurs.

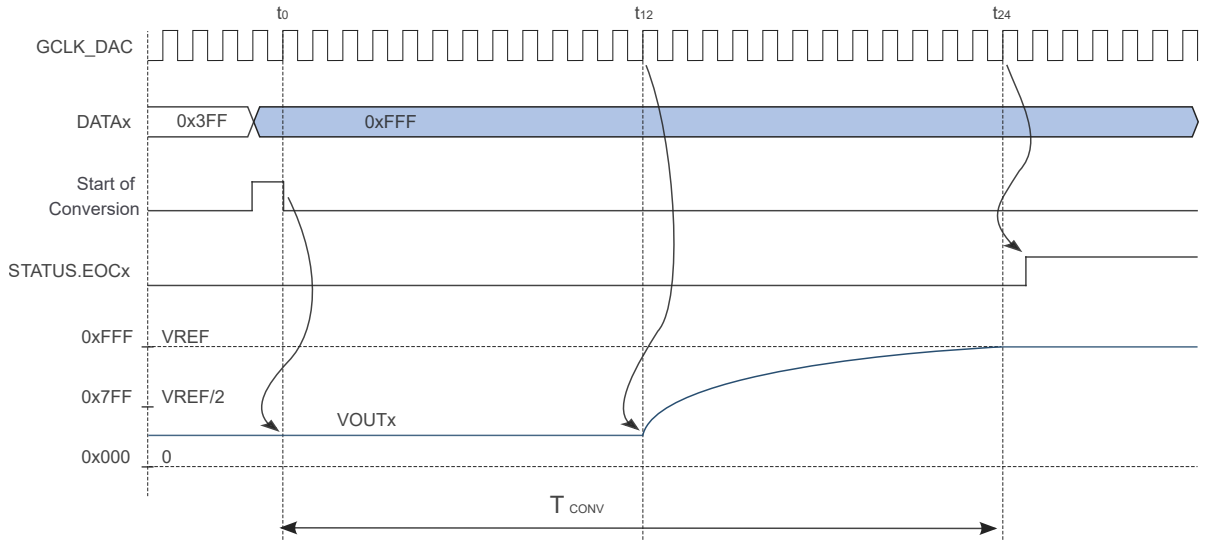
Refer to Events for details. Even if both DAC use the same GCLK, each data conversion can be started independently.

The conversion time is given by the period $T_{GCLK}$ of the generic clock GCLK_DAC and the number of bits:

$$T_{CONV} = 12 \times 2 \times T_{GCLK}$$

The End Of Conversion bit in the Status register indicates that a conversion is completed (STATUS.EOCx=1). This means that VOUTx is stable.

**Figure 47-2.** Single DAC Conversion



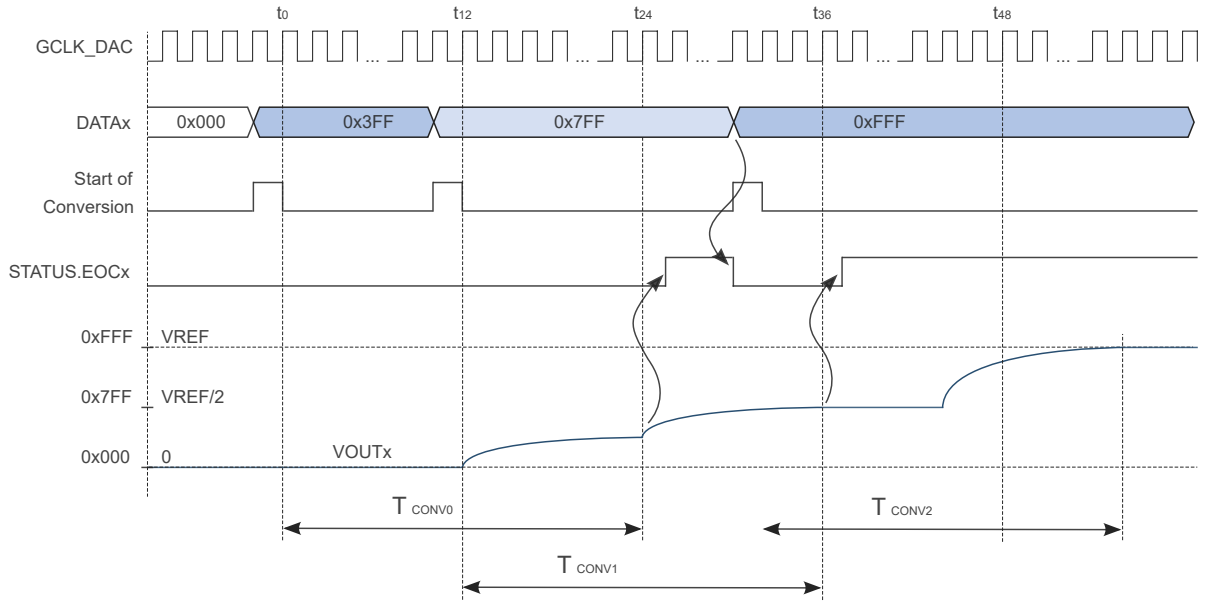

Since the DAC conversion is implemented as pipelined procedure, a new conversion can be started after only 12 GCLK_DAC periods. Therefore if DATAx is written while a conversion is ongoing, start of conversion is postponed until DACx is ready to start next conversion.

The maximum conversion rate (samples per second) is therefore:

$$CR_{max} = \frac{2}{T_{conv}}$$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 47-3.** Multiple DAC Conversions




**Related Links**

SUPC – Supply Controller

### 47.6.3. Operating Conditions

- The DAC voltage reference must be below VDDANA.
- The maximum conversion rate of 1MSPS can be achieved only if VDDANA is above 2.4V.
- The frequency of GCLK_DAC must be equal or lower than 12MHz (corresponding to 1MSPS).

### 47.6.4. DMA Operation

In single mode (CTRLB.DIFF=0), DAC Controller generates the following DMA requests:

- Data Buffer 0 Empty (EMPTY0): The request is set when data is transferred from DATABUF0 or DATA0 to the internal data buffer of DAC0. The request is cleared when either DATA0 register or DATABUF0 register is written, or by writing a '1' to the EMPTY0 bit in the Interrupt Flag register (INTFLAG.EMPTY0).
- Data Buffer 1 Empty (EMPTY1): The request is set when data is transferred from DATABUF1 or DATA1 to the internal data buffer of DAC1. The request is cleared when either DATA0 register or DATABUF1 register is written, or by writing a one to the EMPTY1 bit in the Interrupt Flag register (INTFLAG.EMPTY1).
- Filter 0 Result Ready (RESRDY0): The request is set when the filter is used as standalone, and filter output is ready. The request is cleared by writing a '1' to the RESRDY0 bit in the Interrupt Flag register (INTFLAG.RESRDY0).
- Filter 1 Result Ready (RESRDY1): The request is set when the filter is used as standalone, and filter output is ready. The request is cleared by writing a '1' to the RESRDY1 bit in the Interrupt Flag register (INTFLAG.RESRDY1).

In differential mode (CTRLB.DIFF=1), DAC Controller generates the following DMA request:

- Data Buffer 0 Empty (EMPTY0): The request is set when data is transferred from DATABUF0 or DATA0 to the internal data buffer of DAC1. The request is cleared when either DATA0 register or DATABUF0 register is written, or by writing a one to the EMPTY0 bit in the Interrupt Flag register (INTFLAG.EMPTY0).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

If the CPU accesses the registers which are source of DMA request set/clear condition, the DMA request can be lost or the DMA transfer can be corrupted, if enabled.

### 47.6.5. Interrupts

The DAC Controller has the following interrupt sources:

- DAC0 Data Buffer Empty (EMPTY0): Indicates that the internal data buffer of DAC0 is empty.
- DAC1 Data Buffer Empty (EMPTY1): Indicates that the internal data buffer of DAC1 is empty.
- DAC0 Underrun (UNDERRUN0): Indicates that the internal data buffer of DAC0 is empty and a DAC0 start of conversion event occurred. Refer to Events for details.
- DAC1 Underrun (UNDERRUN1): Indicates that the internal data buffer of DAC1 is empty and a DAC1 start of conversion event occurred. Refer to Events for details.
- Filter 0 Result Ready (RESRDY0): Indicates that Filter 0 result is ready if set as standalone filter.
- Filter 1 Result Ready (RESRDY1): Indicates that Filter 1 result is ready if set as standalone filter.
- Filter 0 Overrun (OVERRUN0): Indicates that the DMA request has not been cleared while the RESULT0 register gets new data.
- Filter 1 Overrun (OVERRUN1): Indicates that the DMA request has not been cleared while the RESULT1 register gets new data.

These interrupts are asynchronous wake-up sources.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register.

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the DAC Controller is reset. See INTFLAG for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated.

### 47.6.6. Events

The DAC Controller can generate the following output events:

- Data Buffer 0 Empty (EMPTY0): Generated when the internal data buffer of DAC0 is empty. Refer to DMA Operation for details.
- Data Buffer 1 Empty (EMPTY1): Generated when the internal data buffer of DAC1 is empty. Refer to DMA Operation for details.
- Filter 0 Result Ready (RESRDY0): Generated when standalone filter 0 result is ready.
- Filter 1 Result Ready (RESRDY1): Generated when standalone filter 1 result is ready.

Writing a '1' to an Event Output bit in the Event Control Register (EVCTRL.EMPTYEOx) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event. Refer to the Event System chapter for details on configuring the event system.

The DAC Controller can take the following actions on an input event:

- DAC0 Start Conversion (START0): DATABUF0 value is transferred into DATA0 as soon as DAC0 is ready for the next conversion, and then conversion is started. START0 is considered as


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to Digital to Analog Conversion for details.

- DAC1 Start Conversion (START1): DATABUF1 value is transferred into DATA1 as soon as DAC1 is ready for the next conversion, and then conversion is started. START1 is considered as asynchronous to GCLK_DAC, thus it is resynchronized in the DAC Controller. Refer to Digital to Analog Conversion for details.
  **Note:** When a DACx Start Conversion event is enabled, only DATABUFx must be written (not DATAx).

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.STARTEIx) enables the corresponding action on input event. Writing a '0' to this bit will disable the corresponding action on input event.
**Note:** When several events are connected to the DAC Controller, the enabled action will be taken on any of the incoming events.

By default, DAC Controller detects rising edge events. Falling edge detection can be enabled by writing '1' to EVCTRL.INVEIx.

Note that if an event occurs before startup time is completed, DATAx is loaded but start of conversion is ignored.

### 47.6.7. Sleep Mode Operation

If the Run In Standby bit in the DAC Control x register DACCCTRLx.RUNSTDBY=1, the DACx will continue the conversions in standby sleep mode.

If DACCCTRLx.RUNSTDBY=0, the DACx will stop conversions in standby sleep mode.

If DACx conversion is stopped in standby sleep mode, DACx is also disabled to reduce power consumption. When exiting standby sleep mode, DACx is enabled again, therefore a certain startup time is required before starting a new conversion.

DAC Controller is compatible with SleepWalking: if RUNSTDBY=1, when an input event (STARTx) is detected in sleep mode, the DAC Controller will request GCLK_DAC in order to complete the conversion.

### 47.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

An exception is the Channel Enable bit in the Peripheral Channel Control registers (PCHCTRLm.CHEN). When changing this bit, the bit value must be read-back to ensure the synchronization is complete and to assert glitch free internal operation. Note that changing the bit value under ongoing synchronization will *not* generate an error.

The following bits are synchronized when written:

- Software Reset bit in control register (CTRLA.SWRST)
- Enable bit in control register (CTRLA.ENABLE)

The following registers are synchronized when written:

- DAC0 data register (DATA0)
- DAC1 data register (DATA1)
- DAC0 data buffer register (DATABUF0)
- DAC1 data buffer register (DATABUF1)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Register Synchronization

## 47.6.9. Additional Features

### 47.6.9.1. DAC0 as Internal Input

The analog output of DAC0, VOUT0, is internally available as input signal for other peripherals (AC and ADC) when DAC0 is enabled.

**Note:** The pin VOUT0 will be dedicated as internal input and cannot be configured as alternate function.

### 47.6.9.2. Output Buffer Current Control

Power consumption can be reduced by controlling the output buffer current, according to conversion rate. Writing to the Current Control bits in DAC Control x register (DACCTRLx.[1:0]) will select an output buffer current.

**Related Links**
DACCTRL0
DAC0 Control
DACCTRL1
DAC1 Control

### 47.6.9.3. Conversion Refresh

Conversion Refresh only works when the input data is not interpolated, i.e. the Oversampling Rate in the DAC Control register is zero (DACCTRLx.OSR=0x0).

The DAC can only maintain its output within one LSB of the desired value for approximately 100µs. When a DAC is used to generate a static voltage or at a rate less than 20kSPS, the conversion must be refreshed periodically. The OSCULP32K clock can start new conversions automatically after a specified period. Write a value to the Refresh bit field in the DAC Control x register (DACCTRLx.REFRESH[3:0]) to select the refresh period according to the formula:

$$T_{\mathrm{REFRESH}} = \mathrm{REFRESH} \times T_{\mathrm{OSCULP32K}}$$

The actual period will depend on the tolerance of the OSCULP32K (see Electrical Characteristics).

If DACCTRLx.REFRESH=0, there is no conversion refresh. DACCTRLx.REFRESH=1 is Reserved.

If no new conversion is started before the refresh period is completed, DACx will convert the DATAx value again.

In standby sleep mode, the refresh mode remains enabled if DACCTRLx.RUNSTDBY=1.

If DATAx is written while a refresh conversion is ongoing, the conversion of the new content of DATAx is postponed until DACx is ready to start the next conversion.

### 47.6.9.4. Differential Mode

DAC0 and DAC1 can be configured to operate in differential mode, i.e. the combined output is a voltage balanced around VREF/2, see also the figure below.

In differential mode, DAC0 and DAC1 are converting synchronously the DATA0 value. DATA0 must therefore be a signed value, represented in two's complement format with DATA0[11] as the signed bit. DATA0 has therefore the range [-2047:2047].

VOUT0 is the positive output and VOUT1 the negative output. The differential output voltage is therefore:

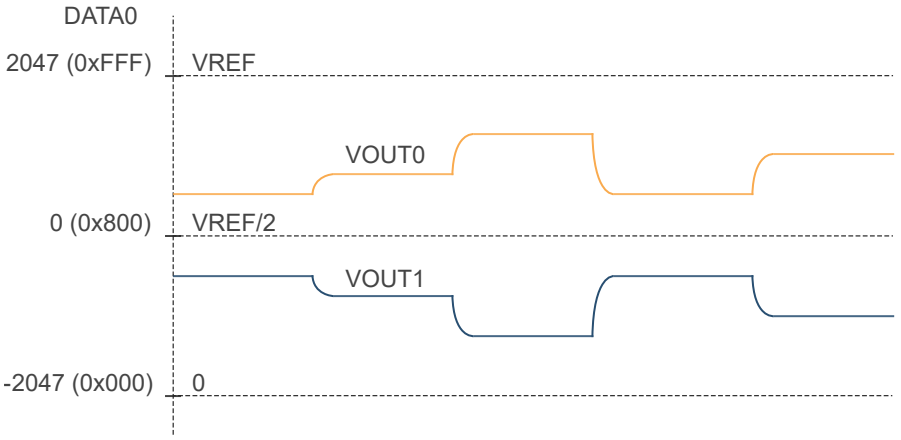
$$V_{\mathrm{OUT}} = \frac{\mathrm{DATA0}}{2047} \times \mathrm{VREF} = (V_{\mathrm{OUT0}} - V_{\mathrm{OUT1}})$$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

DACCTRL0 serves as the configuration register for both DAC0 and DAC1. Therefore DACCTRL1 does not need to be written.

The differential mode is enabled by writing a '1' to the Differential bit in the Control B register (CTRLB.DIFF).

**Figure 47-4.** DAC Conversions in Differential Mode




### 47.6.9.5. Dithering Mode

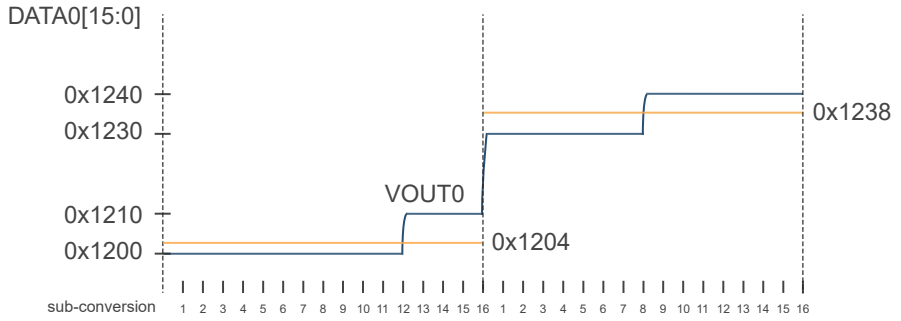
Dithering is enabled by setting DACCTRLx.DITHER to 1. In dithering mode, DATAx is a 16-bit unsigned value where DATAx[15:4] is the 12-bit data converted by DAC and DATAx[3:0] represent the dither bits, used to minimize the quantization error.

The principle is to make 16 sub-conversions of the DATAx[15:4] value or the (DATAx[15:4] + 1) value, so that by averaging those two values, the conversion result of the 16-bit value (DATAx[15:0]) is accurate.

To operate, the STARTx event must be configured to generate 16 events for each DATAx[15:0] conversion, and DATABUFx must be loaded every 16 DAC conversions. EMPTYx event and DMA request are therefore generated every 16 DATABUFx to DATAx transfer. STATUS.EOCx still reports end of each sub-conversions.

Following timing diagram shows examples with DATA0[15:0] = 0x1204 followed by DATA0[15:0] = 0x1238.

**Figure 47-5.** DAC Conversions in Dithering Mode




### 47.6.9.6. Interpolation Mode

The DAC provides interpolation that allows for oversampling ratios (OSR) of 2x, 4x, 8x, 16x or 32x. Interpolation mode is selected by writing a non-zero value to the Oversampling Ratio bits in the DACx Control register (DACCTRLx.OSR).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The data is sampled once over OSR trigger events and then recomputed at the trigger sample rate using a third-order SINC filter.

The figures below show the spectral mask of the SINC filter depending on the selected OSR. $f_s$ is the sampling frequency of the input signal which corresponds to the trigger frequency divided by OSR.

The Filter usage bit DACCTRLx.FEXT determines whether the filter is integrated to the corresponding DAC or used as a standalone filter driven by DMA. If DACCTRLx.FEXT=0, the DAC takes the filter output while the value of RESULTx is reading zero. Conversely, If DACCTRLx.FEXT=1, the DAC value remains zero, and the value of RESULTx register reflects the filter output.

**Figure 47-6.** Interpolator Spectral Mask for 2x OSR

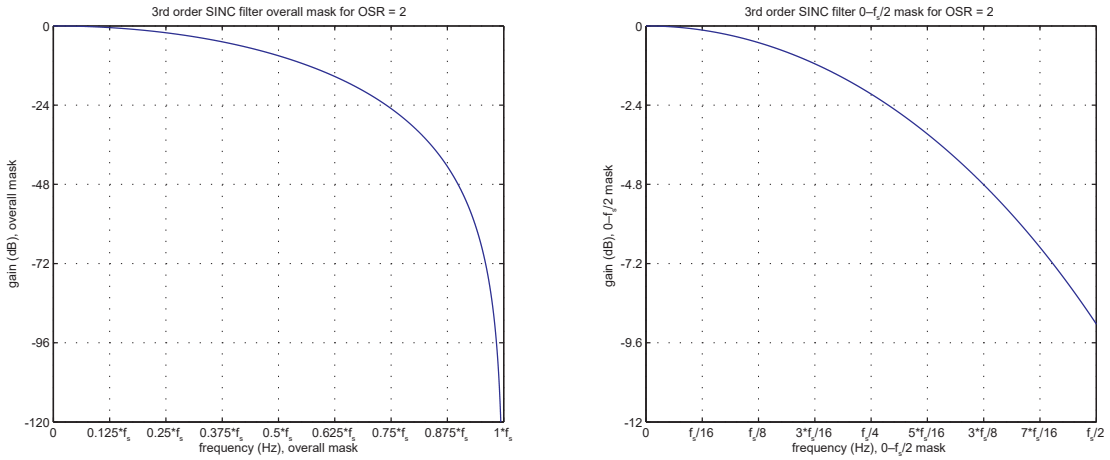
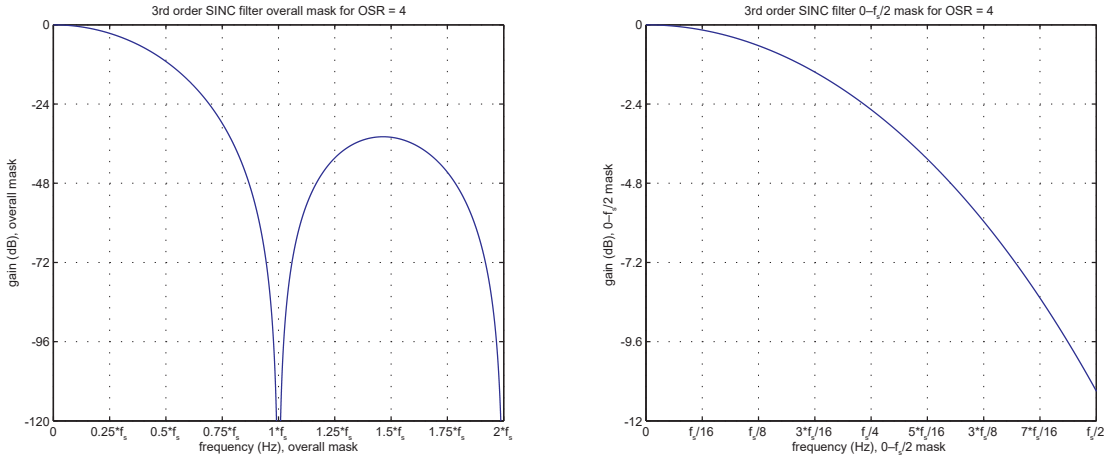



**Figure 47-7.** Interpolator Spectral Mask for 4x OSR





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 47-8.** Interpolator Spectral Mask for 8x OSR

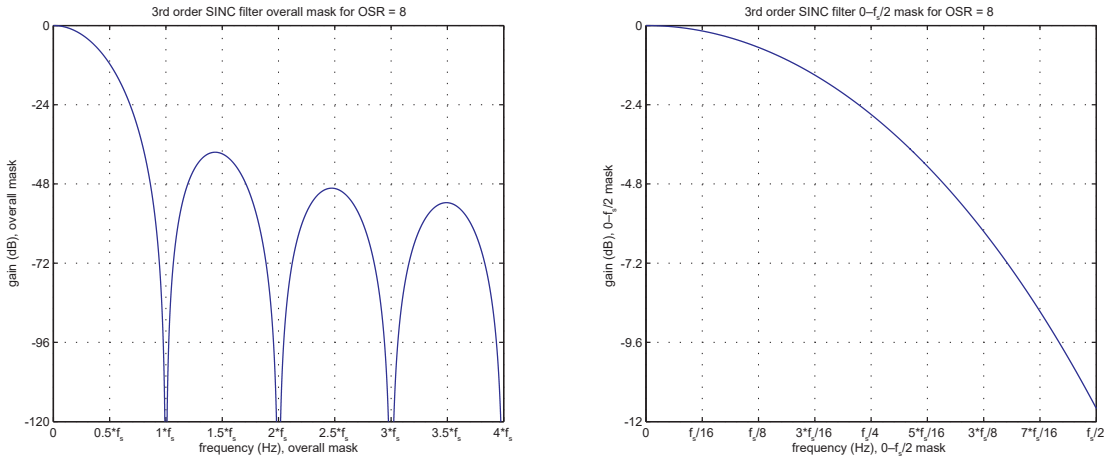



**Figure 47-9.** Interpolator Spectral Mask for 16x OSR

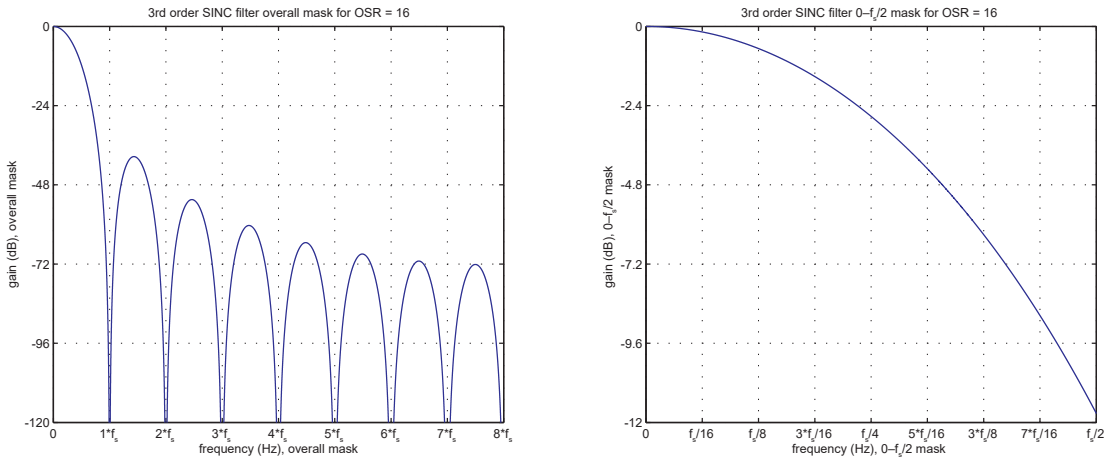
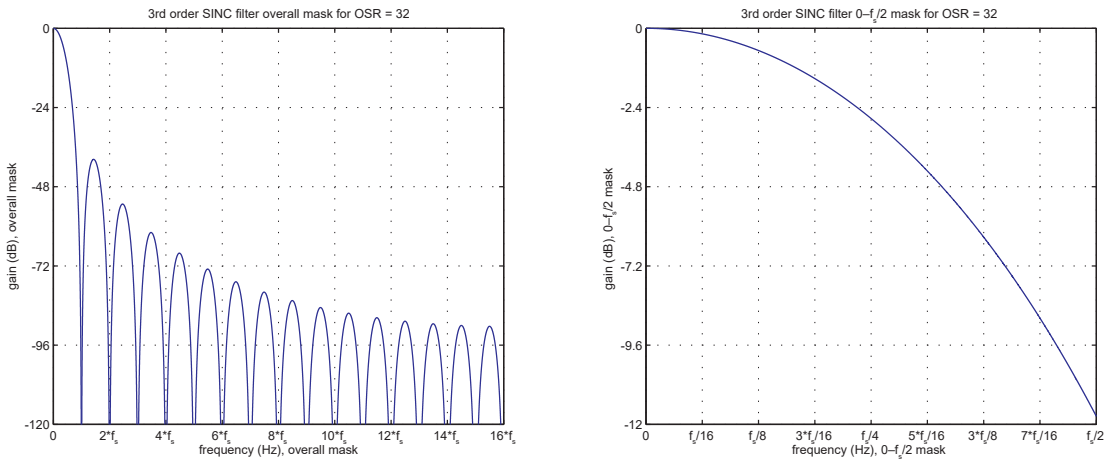



**Figure 47-10.** Interpolator Spectral Mask for 32x OSR



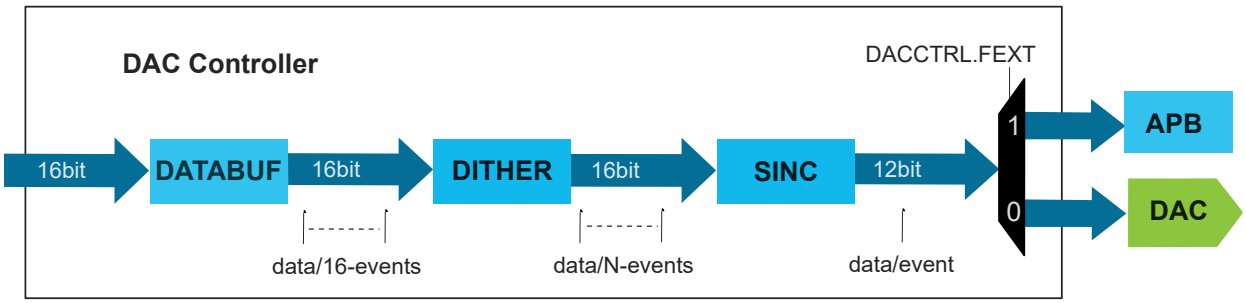

### 47.6.9.7. Dithering-Interpolation Mode

It is possible to enable both Dithering and Interpolation at the same time by setting DACCTRLx.DITHER and DACCTRLx.OSR prior to enabling the DAC. In Dithering-Interpolation mode,


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

the output of dithering is sampled at a number of events corresponding to the OSR value. The valid OSR value is 2, 4, 8, or 16.

**Figure 47-11.** Dithering-Interpolation Data Path





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 47.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | | ENABLE | SWRST |
| 0x01 | CTRLB | 7:0 | | | | | | REFSEL[1:0] | | DIFF |
| 0x02 | EVCTRL | 7:0 | RESRDYEO1 | RESRDYEO0 | INVEI1 | INVEI0 | EMPTYEO1 | EMPTYEO0 | STARTEI1 | STARTEI0 |
| 0x03 | Reserved | | | | | | | | | |
| 0x04 | INTENCLR | 7:0 | OVERRUN1 | OVERRUN0 | RESRDY1 | RESRDY0 | EMPTY1 | EMPTY0 | UNDERRUN1 | UNDERRUN0 |
| 0x05 | INTENSET | 7:0 | OVERRUN1 | OVERRUN0 | RESRDY1 | RESRDY0 | EMPTY1 | EMPTY0 | UNDERRUN1 | UNDERRUN0 |
| 0x06 | INTFLAG | 7:0 | OVERRUN1 | OVERRUN0 | RESRDY1 | RESRDY0 | EMPTY1 | EMPTY0 | UNDERRUN1 | UNDERRUN0 |
| 0x07 | STATUS | 7:0 | | | | | EOC1 | EOC0 | READY1 | READY0 |
| 0x08 | SYNCBUSY | 7:0 | | | DATABUF1 | DATABUF0 | DATA1 | DATA0 | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | DACCTRL0 | 7:0 | DITHER | RUNSTDBY | FEXT | | CCTRL[1:0] | | ENABLE | LEFTADJ |
| | | 15:8 | | OSR[2:0] | | | REFRESH[3:0] | | | |
| 0x0E | DACCTRL1 | 7:0 | DITHER | RUNSTDBY | FEXT | | CCTRL[1:0] | | ENABLE | LEFTADJ |
| | | 15:8 | | OSR[2:0] | | | REFRESH[3:0] | | | |
| 0x10 | DATA0 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| 0x12 | DATA1 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| 0x14 | DATABUF0 | 7:0 | DATABUF[7:0] | | | | | | | |
| | | 15:8 | DATABUF[15:8] | | | | | | | |
| 0x16 | DATABUF1 | 7:0 | DATABUF[7:0] | | | | | | | |
| | | 15:8 | DATABUF[15:8] | | | | | | | |
| 0x18 | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x19 ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | RESULT0 | 7:0 | RESULT[7:0] | | | | | | | |
| | | 15:8 | RESULT[15:8] | | | | | | | |
| 0x1E | RESULT1 | 7:0 | RESULT[7:0] | | | | | | | |
| | | 15:8 | RESULT[15:8] | | | | | | | |

## 47.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – ENABLE** Enable DAC Controller
Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the corresponding bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing '0' to this bit has no effect.
Writing '1' to this bit resets all registers in the DAC to their initial state, and the DAC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
Due to synchronization there is a delay from writing CTRLA.SWRST until the reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the reset is complete.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.2. Control B

**Name:** CTRLB
**Offset:** 0x01
**Reset:** 0x02
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | REFSEL[1:0] | | DIFF |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 1 | 0 |

**Bits 2:1 – REFSEL[1:0]** Reference Selection
This bit field selects the Reference Voltage for both DACs.

| Value | Name | Description |
|---|---|---|
| 0x0 | VREFAU | Unbuffered external voltage reference (not buffered in DAC, direct connection) |
| 0x1 | VDDANA | Voltage supply |
| 0x2 | VREFAB | Buffered external voltage reference (buffered in DAC). This buffered external voltage reference option can be used when the applied external input reference voltage has insufficient drive current. |
| 0x3 | INTREF | Internal bandgap reference |

**Bit 0 – DIFF** Differential Mode Enable
This bit defines the conversion mode for both DACs.

| Value | Description |
|---|---|
| 0 | Single mode |
| 1 | Differential mode |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.3. Event Control

**Name:** EVCTRL
**Offset:** 0x02
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RESRDYEO1 | RESRDYEO0 | INVEI1 | INVEI0 | EMPTYEO1 | EMPTYEO0 | STARTEI1 | STARTEI0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – RESRDYEO1** Enable Result Ready of Filter 1 output event
This bit controls whether the RESRDY1 Event is enabled when the interpolated data is ready.

| Value | Description |
|---|---|
| 0 | Interpolated Data Ready Event is disabled |
| 1 | Interpolated Data Ready Event is enabled |

**Bit 6 – RESRDYEO0** Enable Result Ready of Filter 0 output event
This bit controls whether the RESRDY0 Event is enabled when the interpolated data is ready.

| Value | Description |
|---|---|
| 0 | Interpolated Data Ready Event is disabled |
| 1 | Interpolated Data Ready Event is enabled |

**Bit 5 – INVEI1** Enable Inversion of DAC1 Start Conversion Input Event
This bit defines the detection of the input event for DAC1 START.

| Value | Description |
|---|---|
| 0 | Input event source is not inverted. |
| 1 | Input event source is inverted. |

**Bit 4 – INVEI0** Enable Inversion of DAC0 Start Conversion Input Event
This bit defines the detection of the input event for DAC0 START.

| Value | Description |
|---|---|
| 0 | Input event source is not inverted. |
| 1 | Input event source is inverted. |

**Bit 3 – EMPTYEO1** Data Buffer Empty Event Output DAC1
This bit indicates if the Data Buffer Empty Event output for DAC1 is enabled.

| Value | Description |
|---|---|
| 0 | Data Buffer Empty event is disabled. |
| 1 | Data Buffer Empty event is enabled. |

**Bit 2 – EMPTYEO0** Data Buffer Empty Event Output DAC0
This bit indicates if the Data Buffer Empty Event output for DAC0 is enabled.

| Value | Description |
|---|---|
| 0 | Data Buffer Empty event is disabled. |
| 1 | Data Buffer Empty event is enabled. |

**Bit 1 – STARTEI1** Start Conversion Event Input DAC1
This bit indicates if the Start input event for DAC1 is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | A new conversion will not be triggered on any incoming event.<br>Only DATA1 must be written (not DATABUF1). |
| 1 | A new conversion will be triggered on any incoming event.<br>Only DATABUF1 must be written (not DATA1). |

**Bit 0 – STARTEI0** Start Conversion Event Input DAC0

This bit indicates if the Start input event for DAC0 is enabled.

| Value | Description |
|---|---|
| 0 | A new conversion will not be triggered on any incoming event.<br>Only DATA0 must be written (not DATABUF0). |
| 1 | A new conversion will be triggered on any incoming event.<br>Only DATABUF0 must be written (not DATA0). |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 47.8.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OVERRUN1 | OVERRUN0 | RESRDY1 | RESRDY0 | EMPTY1 | EMPTY0 | UNDERRUN1 | UNDERRUN0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – OVERRUN1** Overrun Interrupt Enable for Filter Channel 1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Interrupt Enable for Filter Channel 1 bit, which disables the Filter 1 Overrun interrupt.

| Value | Description |
|---|---|
| 0 | Filter 1 Result Ready interrupt is disabled. |
| 1 | Filter 1 Result Ready interrupt is enabled. |

**Bit 6 – OVERRUN0** Overrun Interrupt Enable for Filter Channel 0
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun Interrupt Enable for Filter Channel 0 bit, which disables the Filter 0 Overrun interrupt.

| Value | Description |
|---|---|
| 0 | Filter 0 Result Ready interrupt is disabled. |
| 1 | Filter 0 Result Ready interrupt is enabled. |

**Bit 5 – RESRDY1** Filter Channel 1 Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Filter Channel 1 Result Ready Interrupt Enable bit, which disables the Filter Channel 1 Result Ready interrupt.

| Value | Description |
|---|---|
| 0 | Filter 1 Result Ready interrupt is disabled. |
| 1 | Filter 1 Result Ready interrupt is enabled. |

**Bit 4 – RESRDY0** Filter Channel 0 Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Filter Channel 0 Result Ready Interrupt Enable bit, which disables the Filter Channel 0 Result Ready interrupt.

| Value | Description |
|---|---|
| 0 | Filter 0 Result Ready interrupt is disabled. |
| 1 | Filter 0 Result Ready interrupt is enabled. |

**Bit 3 – EMPTY1** Data Buffer 1 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 1 Empty Interrupt Enable bit, which disables the Data Buffer 1 Empty interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The Data Buffer 1 Empty interrupt is disabled. |
| 1 | The Data Buffer 1 Empty interrupt is enabled. |

**Bit 2 – EMPTY0** Data Buffer 0 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 0 Empty Interrupt Enable bit, which disables the Data Buffer 0 Empty interrupt.

| Value | Description |
|---|---|
| 0 | The Data Buffer 0 Empty interrupt is disabled. |
| 1 | The Data Buffer 0 Empty interrupt is enabled. |

**Bit 1 – UNDERRUN1** Underrun Interrupt Enable for DAC1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 1 Underrun Interrupt Enable bit, which disables the Data Buffer 1 Underrun interrupt.

| Value | Description |
|---|---|
| 0 | The Data Buffer 1 Underrun interrupt is disabled. |
| 1 | The Data Buffer 1 Underrun interrupt is enabled. |

**Bit 0 – UNDERRUN0** Underrun Interrupt Enable for DAC0
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 0 Underrun Interrupt Enable bit, which disables the Data Buffer 0 Underrun interrupt.

| Value | Description |
|---|---|
| 0 | The Data Buffer 0 Underrun interrupt is disabled. |
| 1 | The Data Buffer 0 Underrun interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OVERRUN1 | OVERRUN0 | RESRDY1 | RESRDY0 | EMPTY1 | EMPTY0 | UNDERRUN1 | UNDERRUN0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – OVERRUN1** Overrun Interrupt Enable for Filter Channel 1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Interrupt Enable for Filter Channel 1 bit, which enables the Filter 1 Overrun interrupt.

| Value | Description |
|---|---|
| 0 | Filter 1 Result Ready interrupt is disabled. |
| 1 | Filter 1 Result Ready interrupt is enabled. |

**Bit 6 – OVERRUN0** Overrun Interrupt Enable for Filter Channel 0
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overrun Interrupt Enable for Filter Channel 0 bit, which enables the Filter 0 Overrun interrupt.

| Value | Description |
|---|---|
| 0 | Filter 0 Result Ready interrupt is disabled. |
| 1 | Filter 0 Result Ready interrupt is enabled. |

**Bit 5 – RESRDY1** Filter Channel 1 Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Filter Channel 1 Result Ready Interrupt Enable bit, which enables the Filter Channel 1 Result Ready interrupt.

| Value | Description |
|---|---|
| 0 | Filter 1 Result Ready interrupt is disabled. |
| 1 | Filter 1 Result Ready interrupt is enabled. |

**Bit 4 – RESRDY0** Filter Channel 0 Result Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Filter Channel 0 Result Ready Interrupt Enable bit, which enables the Filter Channel 0 Result Ready interrupt.

| Value | Description |
|---|---|
| 0 | Filter 0 Result Ready interrupt is disabled. |
| 1 | Filter 0 Result Ready interrupt is enabled. |

**Bit 3 – EMPTY1** Data Buffer 1 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 1 Empty Interrupt Enable bit, which enables the Data Buffer 1 Empty interrupt.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The Data Buffer 1 Empty interrupt is disabled. |
| 1 | The Data Buffer 1 Empty interrupt is enabled. |

**Bit 2 – EMPTY0** Data Buffer 0 Empty Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 0 Empty Interrupt Enable bit, which enables the Data Buffer 0 Empty interrupt.

| Value | Description |
|---|---|
| 0 | The Data Buffer 0 Empty interrupt is disabled. |
| 1 | The Data Buffer 0 Empty interrupt is enabled. |

**Bit 1 – UNDERRUN1** Underrun Interrupt Enable for DAC1
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 1 Underrun Interrupt Enable bit, which enables the Data Buffer 1 Underrun interrupt.

| Value | Description |
|---|---|
| 0 | The Data Buffer 1 Underrun interrupt is disabled. |
| 1 | The Data Buffer 1 Underrun interrupt is enabled. |

**Bit 0 – UNDERRUN0** Underrun Interrupt Enable for DAC0
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Data Buffer 0 Underrun Interrupt Enable bit, which enables the Data Buffer 0 Underrun interrupt.

| Value | Description |
|---|---|
| 0 | The Data Buffer 0 Underrun interrupt is disabled. |
| 1 | The Data Buffer 0 Underrun interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x06
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | OVERRUN1 | OVERRUN0 | RESRDY1 | RESRDY0 | EMPTY1 | EMPTY0 | UNDERRUN1 | UNDERRUN0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – OVERRUN1** Overrun for Filter Channel 1
This flag is set when the DMA is not cleared while the RESULT1 register gets new data.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun for Filter Channel 0 flag.

| Value | Description |
|---|---|
| 0 | Filter 1 Result Ready interrupt is disabled. |
| 1 | Filter 1 Result Ready interrupt is enabled. |

**Bit 6 – OVERRUN0** Overrun for Filter Channel 0
This flag is set when the DMA is not cleared while the RESULT0 register gets new data.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overrun for Filter Channel 0 flag.

**Bit 5 – RESRDY1** Filter Channel 1 Result Ready
This flag is set when the filter is used as standalone (DACCTRL1.FEXT=1) and the filter output is ready.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Filter Channel 1 Result Ready flag.

**Bit 4 – RESRDY0** Filter Channel 0 Result Ready
This flag is set when the filter is used as standalone (DACCTRL0.FEXT=1) and the filter output is ready.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Filter Channel 0 Result Ready flag.

**Bit 3 – EMPTY1** Data Buffer 1 Empty
This flag is cleared by writing a '1' to it or by writing new data to DATA1 or DATABUF1.
This flag is set when the data buffer for DAC1 is empty and will generate an interrupt request if INTENCLR/INTENSET.EMPTY1=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 1 Empty interrupt flag.

**Bit 2 – EMPTY0** Data Buffer 0 Empty
This flag is cleared by writing a '1' to it or by writing new data to DATA0 or DATABUF0.
This flag is set when the data buffer for DAC0 is empty and will generate an interrupt request if INTENCLR/INTENSET.EMPTY0=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Data Buffer 0 Empty interrupt flag.

**Bit 1 – UNDERRUN1** DAC1 Underrun
This flag is cleared by writing a '1' to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This flag is set when a start conversion event (START1) occurred before new data is copied/written to the DAC1 data buffer and will generate an interrupt request if INTENCLR/INTENSET.UNDERRUN1=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the DAC1 Underrun interrupt flag.

**Bit 0 – UNDERRUN0** DAC0 Underrun

This flag is cleared by writing a '1' to it.
This flag is set when a start conversion event (START0) occurred before new data is copied/written to the DAC) data buffer and will generate an interrupt request if INTENCLR/INTENSET.UNDERRUN0=1.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the DAC0 Underrun interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.7. Status

**Name:** STATUS
**Offset:** 0x07
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | EOC1 | EOC0 | READY1 | READY0 |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 3 – EOC1** DAC1 End of Conversion
This bit is cleared when DATA1 register is written.

| Value | Description |
|---|---|
| 0 | No conversion completed since last load of DATA1. |
| 1 | DAC1 conversion is complete, VOUT1 is stable. |

**Bit 2 – EOC0** DAC0 End of Conversion
This bit is cleared when DATA0 register is written.

| Value | Description |
|---|---|
| 0 | No conversion completed since last load of DATA0. |
| 1 | DAC0 conversion is complete, VOUT0 is stable. |

**Bit 1 – READY1** DAC1 Startup Ready

| Value | Description |
|---|---|
| 0 | DAC1 is not ready for conversion. |
| 1 | Startup time has elapsed, DAC1 is ready for conversion. |

**Bit 0 – READY0** DAC0 Startup Ready

| Value | Description |
|---|---|
| 0 | DAC0 is not ready for conversion. |
| 1 | Startup time has elapsed, DAC0 is ready for conversion. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 47.8.8. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | DATABUF1 | DATABUF0 | DATA1 | DATA0 | ENABLE | SWRST |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 5 – DATABUF1** Data Buffer DAC1
This bit is set when DATABUF1 register is written.
This bit is cleared when DATABUF1 synchronization is completed.

| Value | Description |
|---|---|
| 0 | No ongoing synchronized access. |
| 1 | Synchronized access is ongoing. |

**Bit 4 – DATABUF0** Data Buffer DAC0
This bit is set when DATABUF0 register is written.
This bit is cleared when DATABUF0 synchronization is completed.

| Value | Description |
|---|---|
| 0 | No ongoing synchronized access. |
| 1 | Synchronized access is ongoing. |

**Bit 3 – DATA1** Data DAC1
This bit is set when DATA1 register is written.
This bit is cleared when DATA1 synchronization is completed.

| Value | Description |
|---|---|
| 0 | No ongoing synchronized access. |
| 1 | Synchronized access is ongoing. |

**Bit 2 – DATA0** Data DAC0
This bit is set when DATA0 register is written.
This bit is cleared when DATA0 synchronization is completed.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | No ongoing synchronized access. |
| 1 | Synchronized access is ongoing. |

**Bit 1 – ENABLE** DAC Enable Status

This bit is set when CTRLA.ENABLE bit is written.
This bit is cleared when CTRLA.ENABLE synchronization is completed.

| Value | Description |
|---|---|
| 0 | No ongoing synchronization. |
| 1 | Synchronization is ongoing. |

**Bit 0 – SWRST** Software Reset

This bit is set when CTRLA.SWRST bit is written.
This bit is cleared when CTRLA.SWRST synchronization is completed.

| Value | Description |
|---|---|
| 0 | No ongoing synchronization. |
| 1 | Synchronization is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 47.8.9. DAC0 Control

**Name:** DACCTRL0
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enabled-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OSR[2:0] | | | | REFRESH[3:0] | | | |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DITHER | RUNSTDBY | FEXT | | CCTRL[1:0] | | ENABLE | LEFTADJ |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

**Bits 15:13 – OSR[2:0]** Oversampling Ratio
This field defines the oversampling ratio/interpolation depth.

| Value | Name | Description |
|---|---|---|
| 0x0 | OSR_1 | 1x OSR (no interpolation) |
| 0x1 | OSR_2 | 2x OSR |
| 0x2 | OSR_4 | 4x OSR |
| 0x3 | OSR_8 | 8x OSR |
| 0x4 | OSR_16 | 16x OSR |
| 0x5 | OSR_32 | 32x OSR |
| other | - | Reserved |

**Bits 11:8 – REFRESH[3:0]** Refresh period
This field defines the refresh period. If REFRESH=0x0, the refresh mode is disabled. If REFRESH>0x1, else the refresh period is:
$T_{\mathrm{REFRESH}} = \mathrm{REFRESH} \times 30\mu s$

**Bit 7 – DITHER** Dithering Mode

| Value | Description |
|---|---|
| 0 | Dithering mode is disabled. |
| 1 | Dithering mode is enabled. |

**Bit 6 – RUNSTDBY** Run in Standby
This bit controls the behavior of DAC0 during standby sleep mode.

| Value | Description |
|---|---|
| 0 | DAC0 is disabled during standby sleep mode. |
| 1 | DAC0 continues to operate during standby sleep mode. |

**Bit 5 – FEXT** External Filter Enable
This bit controls the usage of the filter.

| Value | Description |
|---|---|
| 0 | The filter is integrated to the DAC |
| 1 | The filter is used as standalone |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 3:2 – CCTRL[1:0]** Current Control

This field defines the current in output buffer according to conversion rate.

**Current Control**

| Value | Name | Description |
|---|---|---|
| 0x0 | CC100K | GCLK_DAC ≤ 1.2MHz (100kSPS) |
| 0x1 | CC1M | 1.2MHz < GCLK_DAC ≤ 6MHz (500kSPS) |
| 0x2 | CC12M | 6MHz < GCLK_DAC ≤ 12MHz (1MSPS) |
| 0x3 | Reserved | |

**Bit 1 – ENABLE** Enable DAC0

This bit enables DAC0 when DAC Controller is enabled (CTRLA.ENABLE).

| Value | Description |
|---|---|
| 0 | DAC0 is disabled. |
| 1 | DAC0 is enabled. |

**Bit 0 – LEFTADJ** Left Adjusted Data

This bit controls how the 12-bit conversion data is adjusted in the Data and Data Buffer registers.

| Value | Description |
|---|---|
| 0 | DATA0 and DATABUF0 registers are right-adjusted. |
| 1 | DATA0 and DATABUF0 registers are left-adjusted. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.10. DAC1 Control

**Name:** DACCTRL1
**Offset:** 0x0E
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enabled-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OSR[2:0] | | | | REFRESH[3:0] | | | |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DITHER | RUNSTDBY | FEXT | | CCTRL[1:0] | | ENABLE | LEFTADJ |
| Access | R/W | R/W | R/W | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | 0 | 0 | 0 | 0 |

**Bits 15:13 – OSR[2:0]** Oversampling Ratio
This field defines the oversampling ratio/interpolation depth.

| Value | Name | Description |
|---|---|---|
| 0x0 | OSR_1 | 1x OSR (no interpolation) |
| 0x1 | OSR_2 | 2x OSR |
| 0x2 | OSR_4 | 4x OSR |
| 0x3 | OSR_8 | 8x OSR |
| 0x4 | OSR_16 | 16x OSR |
| 0x5 | OSR_32 | 32x OSR |
| other | - | Reserved |

**Bits 11:8 – REFRESH[3:0]** Refresh period
This field defines the refresh period. If REFRESH=0x0, the refresh mode is disabled. If REFRESH>0x1, else the refresh period is:
$T_{\mathrm{REFRESH}} = \mathrm{REFRESH} \times 30\mu s$

**Bit 7 – DITHER** Dithering Mode

| Value | Description |
|---|---|
| 0 | Dithering mode is disabled. |
| 1 | Dithering mode is enabled. |

**Bit 6 – RUNSTDBY** Run in Standby
This bit controls the behavior of DAC1 during standby sleep mode.

| Value | Description |
|---|---|
| 0 | DAC1 is disabled during standby sleep mode. |
| 1 | DAC1 continues to operate during standby sleep mode. |

**Bit 5 – FEXT** External Filter Enable
This bit controls the usage of the filter.

| Value | Description |
|---|---|
| 0 | The filter is integrated to the DAC |
| 1 | The filter is used as standalone |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 3:2 – CCTRL[1:0]** Current Control

This field defines the current in output buffer.

**Current Control**

| Value | Name | Description |
|---|---|---|
| 0x0 | CC100K | GCLK_DAC <= 1.2MHz (100kSPS) |
| 0x1 | CC1M | 1.2MHz < GCLK_DAC <= 6MHz (500kSPS) |
| 0x2 | CC12M | 6MHz < GCLK_DAC <= 12MHz (1MSPS) |
| 0x3 | | Reserved |

**Bit 1 – ENABLE** Enable DAC1

This bit enables DAC1 when DAC Controller is enabled (CTRLA.ENABLE).

| Value | Description |
|---|---|
| 0 | DAC1 is disabled. |
| 1 | DAC1 is enabled. |

**Bit 0 – LEFTADJ** Left Adjusted Data

This bit controls how the 12-bit conversion data is adjusted in the Data and Data Buffer registers.

| Value | Description |
|---|---|
| 0 | DATA1 and DATABUF1 registers are right-adjusted. |
| 1 | DATA1 and DATABUF1 registers are left-adjusted. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.11. Data DAC0

**Name:** DATA0
**Offset:** 0x10
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DATA[15:0]** DAC0 Data

DATA0 register contains the 12-bit value that is converted to a voltage by the DAC0. The adjustment of these 12 bits within the 16-bit register is controlled by DACCTRL0.LEFTADJ:
- DATA[11:0] when DACCTRL0.LEFTADJ=0.
- DATA[15:4] when DACCTRL0.LEFTADJ=1.

In dithering mode (whatever DACCTRL0.LEFTADJ value):
- DATA[15:4] are the 12-bit converted by DAC0.
- DATA[3:0] are the dither bits.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.12. Data DAC1

**Name:** DATA1
**Offset:** 0x12
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DATA[15:0]** DAC1 Data

DATA1 register contains the 12-bit value that is converted to a voltage by the DAC1. The adjustment of these 12 bits within the 16-bit register is controlled by DACCTRL1.LEFTADJ:
- DATA[11:0] when DACCTRL1.LEFTADJ=0.
- DATA[15:4] when DACCTRL1.LEFTADJ=1.

In dithering mode (whatever DACCTRL1.LEFTADJ value):
- DATA[15:4] are the 12-bit converted by DAC1.
- DATA[3:0] are the dither bits.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.13. Data Buffer DAC0

**Name:** DATABUF0
**Offset:** 0x14
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATABUF[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATABUF[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DATABUF[15:0]** DAC0 Data Buffer
DATABUF0 contains the value to be transferred into DATA0 when a START0 event occurs.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.14. Data Buffer DAC1

**Name:** DATABUF1
**Offset:** 0x16
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATABUF[15:8] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATABUF[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DATABUF[15:0]** DAC1 Data Buffer
DATABUF1 contains the value to be transferred into DATA1 when a START1 event occurs.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.15. Debug Control

**Name:** DBGCTRL
**Offset:** 0x18
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run
This bit is not reset by a software reset.
This bits controls the functionality when the CPU is halted by an external debugger.

| Value | Description |
|---|---|
| 0 | The DAC is halted when the CPU is halted by an external debugger. Any ongoing conversion will complete. |
| 1 | The DAC continues normal operation when the CPU is halted by an external debugger. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 47.8.16. Result 0

**Name:** RESULT0
**Offset:** 0x1C
**Reset:** 0x0000
**Property:** Read-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RESULT[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RESULT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RESULT[15:0]** Channel 0 Filter Output
RESULT[15:0] contains the value of the interpolated data written to DATA0 or DATABUF0 in standalone mode (DACCTRL0.FEXT=1).


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 47.8.17. Result 1

**Name:** RESULT1
**Offset:** 0x1E
**Reset:** 0x0000
**Property:** Read-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RESULT[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RESULT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – RESULT[15:0]** Channel 0 Filter Output
RESULT[15:0] contains the value of the interpolated data written to DATA1 or DATABUF1 in standalone mode (DACCTRL1.FEXT=1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 48. Timer/Counter (TC)

## 48.1. Overview

There are up to eight TC peripheral instances.

Each TC consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter, together with the compare/capture channels, can be configured to timestamp input events or I/O pin edges, allowing for capturing of frequency and/or pulse width.

A TC can also perform waveform generation, such as frequency generation and pulse-width modulation.

## 48.2. Features

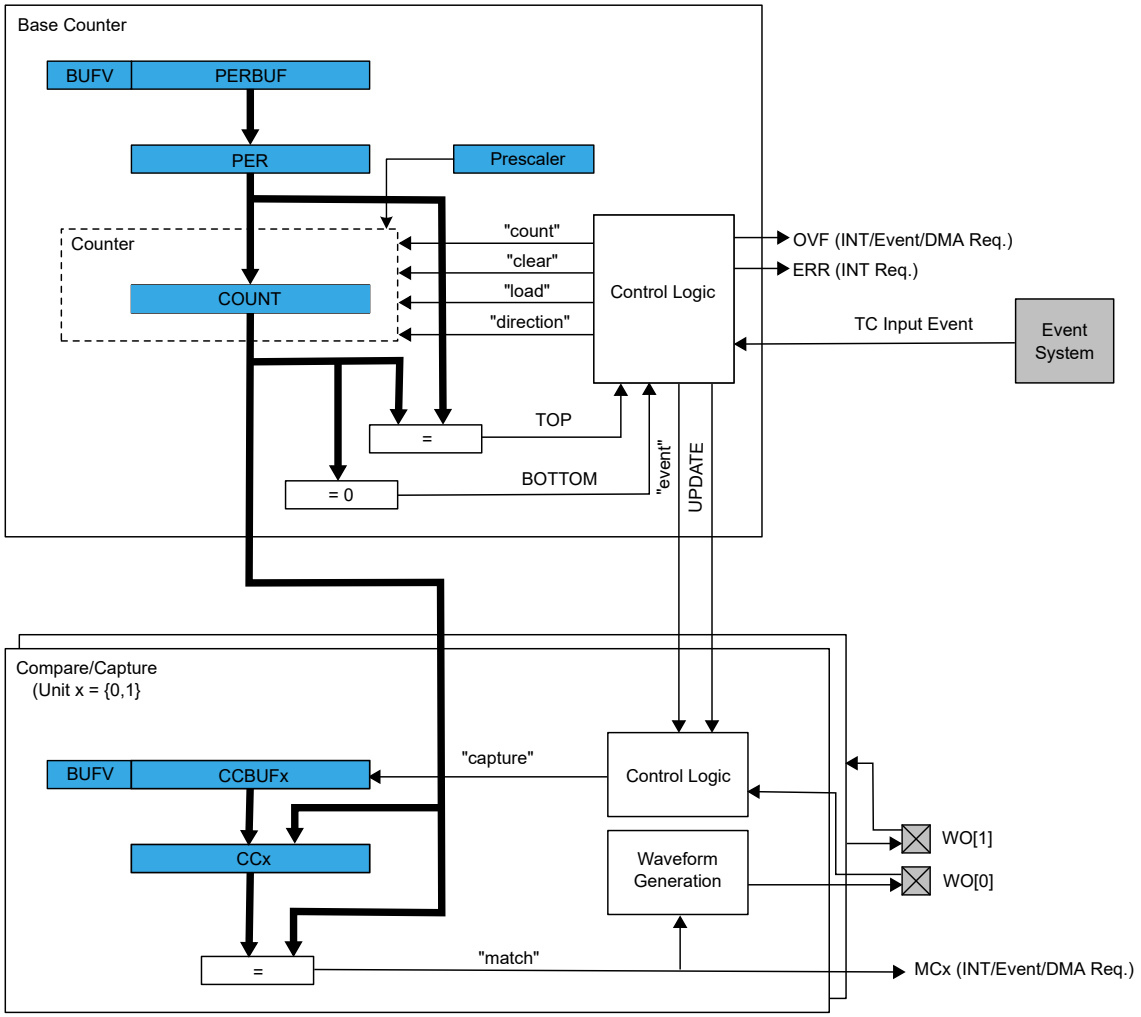
The following are key features of the TC module:

- Selectable configuration:
  - 8-bit, 16-bit, or 32-bit TC operation, with compare/capture channels
- 2 compare/capture channels (CC) with:
  - Double buffered timer period setting (in 8-bit mode only)
  - Double buffered compare channel
- Waveform generation
  - Frequency generation
  - Single-slope Pulse-Width Modulation (PWM)
- Input capture
  - Event or I/O pin edge capture
  - Frequency capture
  - Pulse-width capture
  - Time-stamp capture
  - Minimum and maximum capture
- One input event
- Interrupts/output events on:
  - Counter overflow/underflow
  - Compare match or capture
- Internal prescaler
- DMA support


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48.3. Block Diagram

**Figure 48-1.** Timer/Counter Block Diagram




## 48.4. Signal Description

**Table 48-1.** Signal Description for TC.

| Signal Name | Type | Description |
|---|---|---|
| WO[1:0] | Digital output | Waveform output |
| | Digital input | Capture input |

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

**Related Links**

I/O Multiplexing and Considerations


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 48.5.1. I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

**Table 48-2.** I/O Lines

| Instance | Signal | I/O Line | Peripheral Function |
|---|---|---|---|
| MODULE0 | SIGNAL | PAxx | A |

**Related Links**

PORT - I/O Pin Controller

### 48.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake up the device from Sleep modes. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

**Related Links**

PM – Power Manager

### 48.5.3. Clocks

The TC bus clocks (CLK_TCx_APB) can be enabled and disabled in the Main Clock Module. The default state of CLK_TCx_APB can be found in the *Peripheral Clock Masking*.

The generic clocks (GCLK_TCx) are asynchronous to the user interface clock (CLK_TCx_APB). Due to this asynchronicity, accessing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Note:** Two instances of the TC may share a peripheral clock channel. In this case, they cannot be set to different clock frequencies. Refer to the peripheral clock channel mapping of the Generic Clock Controller (GCLK.PCHTRLm) to identify shared peripheral clocks.

**Related Links**

PCHCTRLm

Peripheral Channel Control

Peripheral Clock Masking

### 48.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

**Related Links**

DMAC – Direct Memory Access Controller

### 48.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.5.6. Events

The events of this peripheral are connected to the Event System.

**Related Links**

Event System (EVSYS)

### 48.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

**Related Links**

DBGCTRL

Debug Control

### 48.5.8. Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)
- Status register (STATUS)
- Count register (COUNT)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture Value registers and Compare/Capture Value Buffer registers (CCx, CCBUFx)

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

### 48.5.9. Analog Connections

Not applicable.

## 48.6. Functional Description

### 48.6.1. Principle of Operation

The following definitions are used throughout the documentation:

**Table 48-3.** Timer/Counter Definitions

| Name | Description |
|---|---|
| TOP | The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the waveform generator mode in Waveform Output Operations. |
| ZERO | The counter is ZERO when it contains all zeroes |
| MAX | The counter reaches MAX when it contains all ones |
| UPDATE | The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings. |
| Timer | The timer/counter clock control is handled by an internal source |
| Counter | The clock control is handled externally (e.g. counting external events) |
| CC | For compare operations, the CC are referred to as “compare channels”<br>For capture operations, the CC are referred to as “capture channels.” |

Each TC instance has up to two compare/capture channels (CC0 and CC1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The counter in the TC can either count events from the Event System, or clock ticks of the GCLK_TCx clock, which may be divided by the prescaler.

The counter value is passed to the CCx where it can be either compared to user-defined values or captured.

For optimized timing the CCx and CCBUFx registers share a common resource. When writing into CCBUFx, lock the access to the corresponding CCx register (SYNCBUSY.CCX = 1) till the CCBUFx register value is not loaded into the CCx register (BUFVx == 1). Each buffer register has a buffer valid (BUFV) flag that indicates when the buffer contains a new value.

The Counter register (COUNT) and the Compare and Capture registers with buffers (CCx and CCBUFx) can be configured as 8-, 16- or 32-bit registers, with according MAX values. Mode settings (CTRLA.MODE) determine the maximum range of the Counter register.

In 8-bit mode, a Period Value (PER) register and its Period Buffer Value (PERBUF) register are also available. The counter range and the operating frequency determine the maximum time resolution achievable with the TC peripheral.

The TC can be set to count up or down. Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached that value. On a comparison match the TC can request DMA transactions, or generate interrupts or events for the Event System.

In compare operation, the counter value is continuously compared to the values in the CCx registers. In case of a match the TC can request DMA transactions, or generate interrupts or events for the Event System. In waveform generator mode, these comparisons are used to set the waveform period or pulse width.

Capture operation can be enabled to perform input signal period and pulse width measurements, or to capture selectable edges from an IO pin or internal event from Event System.

## 48.6.2. Basic Operation

### 48.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the TC is disabled (CTRLA.ENABLE = 0):

- The Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits
- The Drive Control register (DRVCTRL)
- The Wave register (WAVE)
- The Event Control register (EVCTRL)

Writing to Enable-Protected bits and setting the CTRLA.ENABLE bit can be performed in a single 32-bit access of the CTRLA register. Writing to Enable-Protected bits and clearing the CTRLA.ENABLE bit cannot be performed in a single 32-bit access.

Before enabling the TC, follow these steps to configure the peripheral:

1. Enable the TC bus clock (CLK_TCx_APB).
2. Select 8-bit, 16-bit, or 32-bit counter mode through the TC Mode bit group in the Control A register (CTRLA.MODE). The default mode is 16-bit.
3. Select one wave generation operation in the Waveform Generation Operation bit group in the WAVE register (WAVE.WAVEGEN).
4. If desired, the GCLK_TCx clock can be prescaled through the Prescaler bit group in the Control A register (CTRLA.PRESCALER).
   - If the prescaler is used, select a prescaler synchronization operation through the Prescaler and Counter Synchronization bit group in the Control A register (CTRLA.PRESYNC).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

5. If desired, select one-shot operation by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT).
6. If desired, configure the counting direction 'down' (starting from the TOP value) by writing a '1' to the Counter Direction bit in the Control B register (CTRLBSET.DIR).
7. For capture operation, enable the individual channels to capture in the Capture Channel x Enable bit group in the Control A register (CTRLA.CAPTEN).
8. If desired, enable inversion of the waveform output or IO pin input signal for individual channels through the Invert Enable bit group in the Drive Control register (DRVCTRL.INVEN).

#### 48.6.2.2. Enabling, Disabling, and Resetting

The TC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TC is disabled by writing a zero to CTRLA.ENABLE.

The TC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TC, except DBGCTRL, will be reset to their initial state. Refer to the CTRLA register for details.

The TC should be disabled before the TC is reset in order to avoid undefined behavior.

#### 48.6.2.3. Prescaler Selection

The GCLK_TCx is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

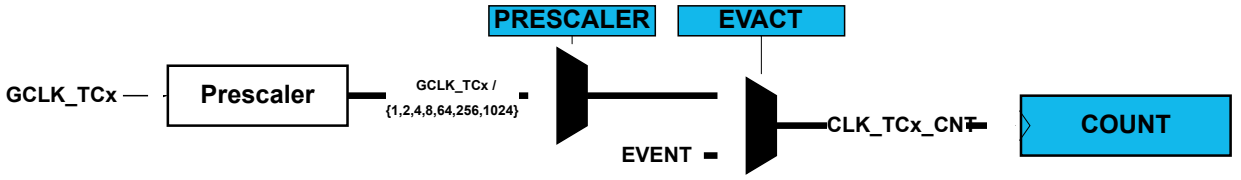
If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCx clock pulse or the next prescaled clock pulse. For further details, refer to Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) description.

Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

**Note:** When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TCx_CNT.

**Figure 48-2.** Prescaler




#### 48.6.2.4. Counter Mode

The counter mode is selected by the Mode bit group in the Control A register (CTRLA.MODE). By default, the counter is enabled in the 16-bit counter resolution. The following three counter resolutions are available:

- COUNT8: The 8-bit TC has its own Period Value and Period Buffer Value registers (PER and PERBUF).
- COUNT16: 16-bit is the default counter mode. There is no dedicated period register in this mode.
- COUNT32: 32-bit mode is achieved by pairing two 16-bit TC peripherals. TC(2n) is paired with TC(2n+1).
  When paired, the TC peripherals are configured using the registers of the even-numbered TC.
  The TC bus clocks (CLK_TCx_APB) for both Host and Client TCs need to be enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The odd-numbered partner will act as a Client, and the Client bit in the Status register (STATUS.Client) will be set. The register values of a Client will not reflect the registers of the 32-bit counter. Writing to any of the Client registers will not affect the 32-bit counter. Normal access to the Client COUNT and CCx registers is not allowed.

### 48.6.2.5. Counter Operations

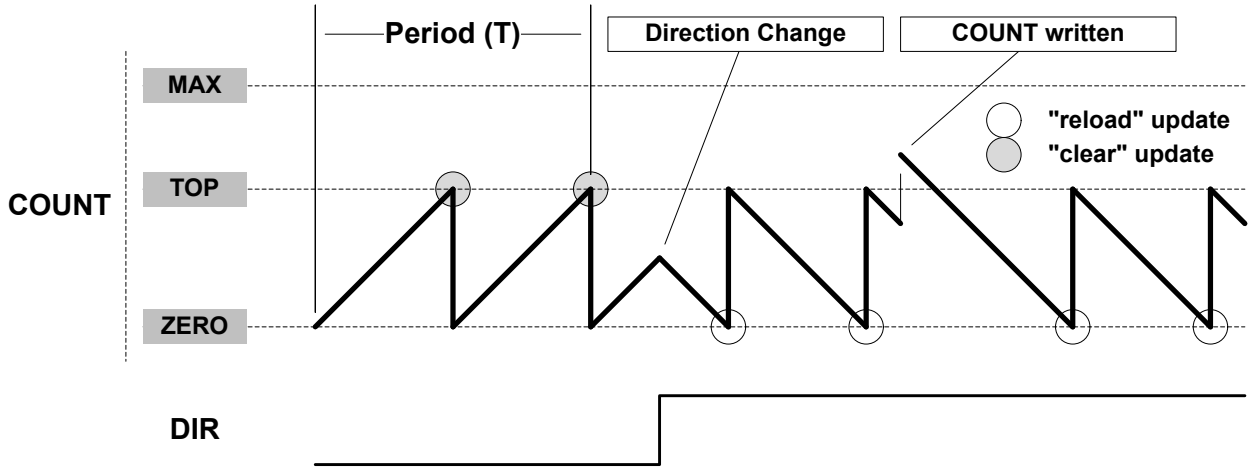
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TC clock input (CLK_TCx_CNT). A counter clear or reload marks the end of the current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If this bit is zero the counter is counting up, and counting down if CTRLB.DIR = 1. The counter will count up or count down for each tick (clock or event) until it reaches TOP or ZERO. When it is counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When it is counting down, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

INTFLAG.OVF can be used to trigger an interrupt, a DMA request, or an event. An overflow/underflow occurrence (i.e., a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT).

It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running. When starting the TC, the COUNT value will be either ZERO or TOP (depending on the counting direction set by CTRLBSET.DIR or CTRLBCLR.DIR), unless a different value has been written to it, or the TC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed when the counter is running, see figure below.

**Figure 48-3.** Counter Operation




Due to asynchronous clock domains, the internal counter settings are written when the synchronization is complete. Normal operation must be used when using the counter as timer base for the capture channels.

#### 48.6.2.5.1. Stop Command and Event Action

A Stop command can be issued from software by using Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP). When a Stop is detected while the counter is running, the counter will not retain its current value. All waveforms are cleared and the Stop bit in the Status register is set (STATUS.STOP).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 48.6.2.5.2. Re-Trigger Command and Event Action

A re-trigger command can be issued from software by writing the Command bits in the Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER), or from event when a re-trigger event action is configured in the Event Control register (EVCTRL.EVACT = 0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). When the re-trigger command is detected while the counter is stopped, the counter will resume counting from the current value in the COUNT register.

**Note:** When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT=0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

#### 48.6.2.5.3. Count Event Action

The TC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR). The count event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x2, COUNT).

**Note:** If this operation mode is selected, PWM generation is not supported.

#### 48.6.2.5.4. Start Event Action

The TC can start counting operation on an event when previously stopped. In this configuration, the event has no effect if the counter is already counting. When the peripheral is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

The Start TC on Event action can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT=0x3, START).

### 48.6.2.6. Compare Operations

By default, the Compare/Capture channel is configured for compare operations.

When using the TC and the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare Buffer (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a forced update command (CTRLBSET.CMD=UPDATE). For further details, refer to Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

#### 48.6.2.6.1. Waveform Output Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a Waveform Generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Output Waveform x Invert Enable bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.
   **Note:** Event must not be used when the compare channel is set in waveform output operating mode.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TCx_CNT (see Normal Frequency Operation). An interrupt/and or event can be generated on comparison match if enabled. The same condition generates a DMA request.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

There are four waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

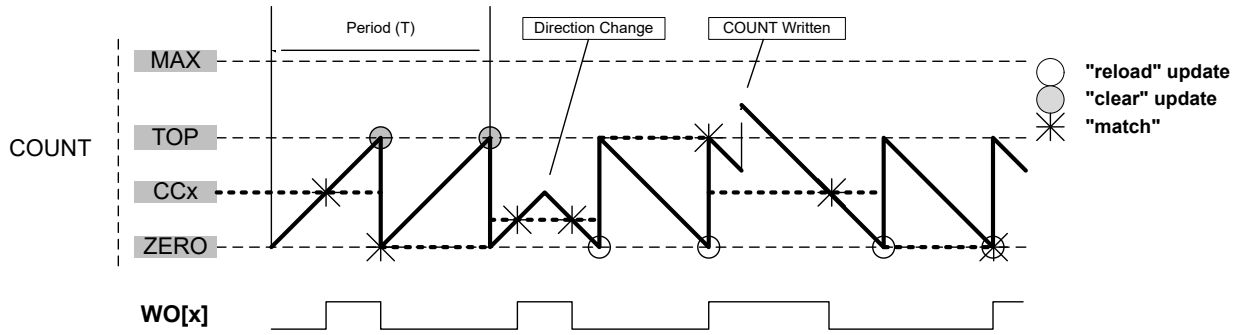
- Normal frequency (NFRQ)
- Match frequency (MFRQ)
- Normal pulse-width modulation (NPWM)
- Match pulse-width modulation (MPWM)

When using NPWM or NFRQ configuration, the TOP will be determined by the counter resolution. In 8-bit Counter mode, the Period register (PER) is used as TOP, and the TOP can be changed by writing to the PER register. In 16- and 32-bit Counter mode, TOP is fixed to the maximum (MAX) value of the counter.

### Normal Frequency Generation (NFRQ)

For Normal Frequency Generation, the period time (T) is controlled by the period register (PER) for 8-bit Counter mode and MAX for 16- and 32-bit mode. The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (INTFLAG.MCx) will be set.

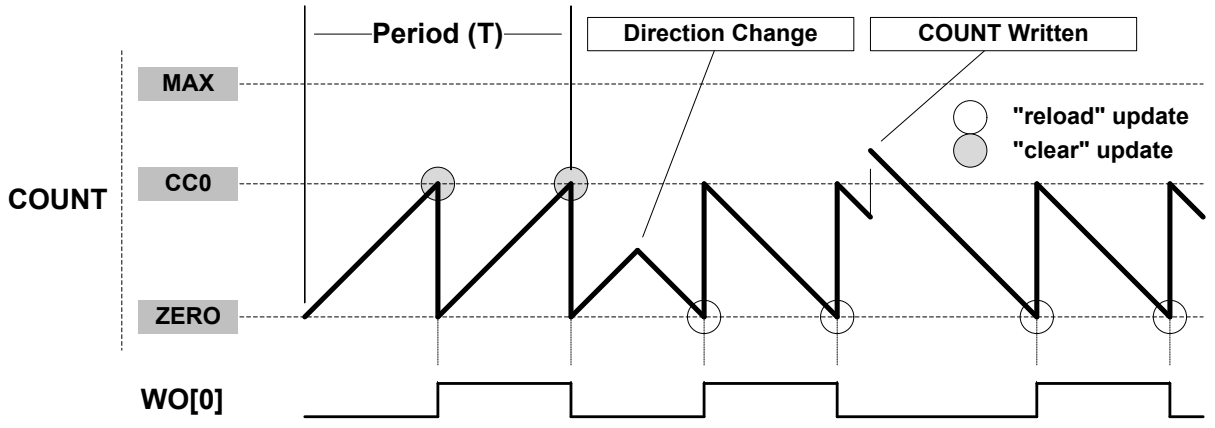
**Figure 48-4.** Normal Frequency Operation




### Match Frequency Generation (MFRQ)

For Match Frequency Generation, the period time (T) is controlled by the CC0 register instead of PER or MAX. WO[0] toggles on each Update condition.

**Figure 48-5.** Match Frequency Operation




### Normal Pulse-Width Modulation Operation (NPWM)

NPWM uses single-slope PWM generation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For single-slope PWM generation, the period time (T) is controlled by the TOP value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.

The following equation calculates the exact resolution for a single-slope PWM ($R_{PWM_SS}$) waveform:

$$R_{PWM_SS} = \frac{\log(TOP+1)}{\log(2)}$$

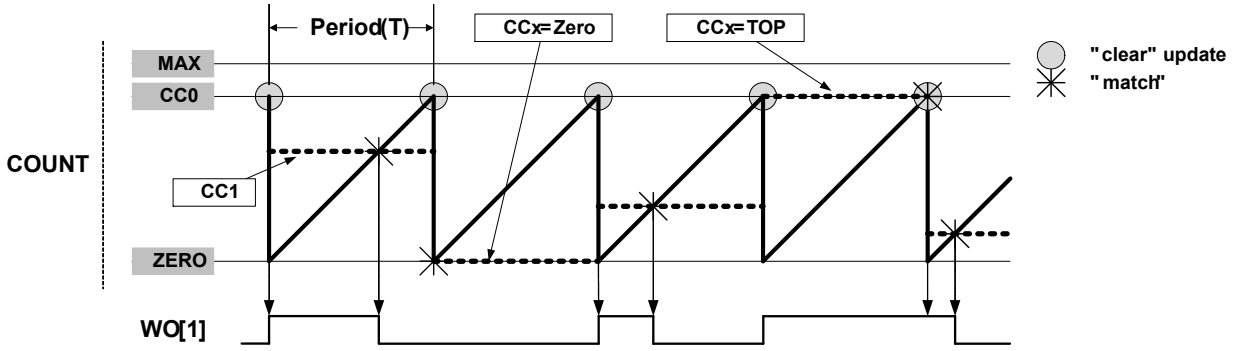
The PWM frequency ($f_{PWM_SS}$) depends on TOP value and the peripheral clock frequency ($f_{GCLK_TCx}$), and can be calculated by the following equation:

$$f_{PWM_SS} = \frac{f_{GCLK_TCx}}{N(TOP+1)}$$

Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

**Match Pulse-Width Modulation Operation (MPWM)**
In MPWM, the output of WO[1] is depending on CC1 as shown in the figure below. On every overflow/underflow, a one-TC-clock-cycle negative pulse is put out on WO[0] (not shown in the figure).

**Figure 48-6.** Match PWM Operation




The table below shows the Update Counter and Overflow Event/Interrupt Generation conditions in different operation modes.

**Table 48-4.** Counter Update and Overflow Event/interrupt Conditions in TC

| Name | Operation | TOP | Update | Output Waveform | | OVFIF/Event | |
|---|---|---|---|---|---|---|---|
| | | | | On Match | On Update | Up | Down |
| NFRQ | Normal Frequency | PER | TOP/ ZERO | Toggle | Stable | TOP | ZERO |
| MFRQ | Match Frequency | CC0 | TOP/ ZERO | Toggle | Stable | TOP | ZERO |
| NPWM | Single-slope PWM | PER | TOP/ ZERO | See description above. | | TOP | ZERO |
| MPWM | Single-slope PWM | CC0 | TOP/ ZERO | Toggle | Toggle | TOP | ZERO |

**Related Links**
PORT - I/O Pin Controller

### 48.6.2.7. Double Buffering

The Compare Channels (CCx) registers, and the Period (PER) register in 8-bit mode are double buffered. Each buffer register has a buffer valid bit (CCBUFVx or PERBUFV) in the STATUS register, which indicates that the buffer register contains a new valid value that can be copied into the

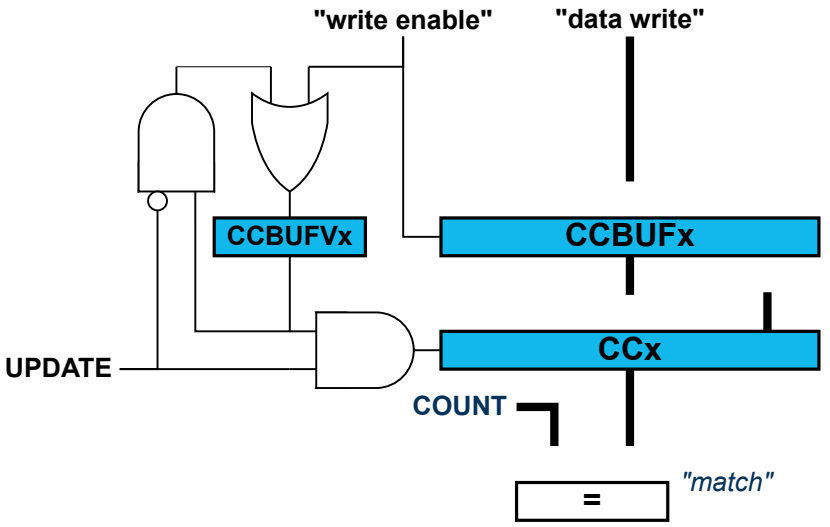

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

corresponding register. As long as the respective buffer valid status flag (PERBUFV or CCBUFVx) are set to '1', related syncbusy bits are set (SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and access to the respective PER or CCx register is invalid.

When the buffer valid flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the buffer valid flags bit in the STATUS register are automatically cleared by hardware.
**Note:** The software update command (CTRLBSET.CMD=0x3) is acting independently of the LUPD value.

A compare register is double buffered as in the following figure.

**Figure 48-7.** Compare Channel Double Buffering




Both the registers (PER/CCx) and corresponding buffer registers (PERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLBSET.LUPD.

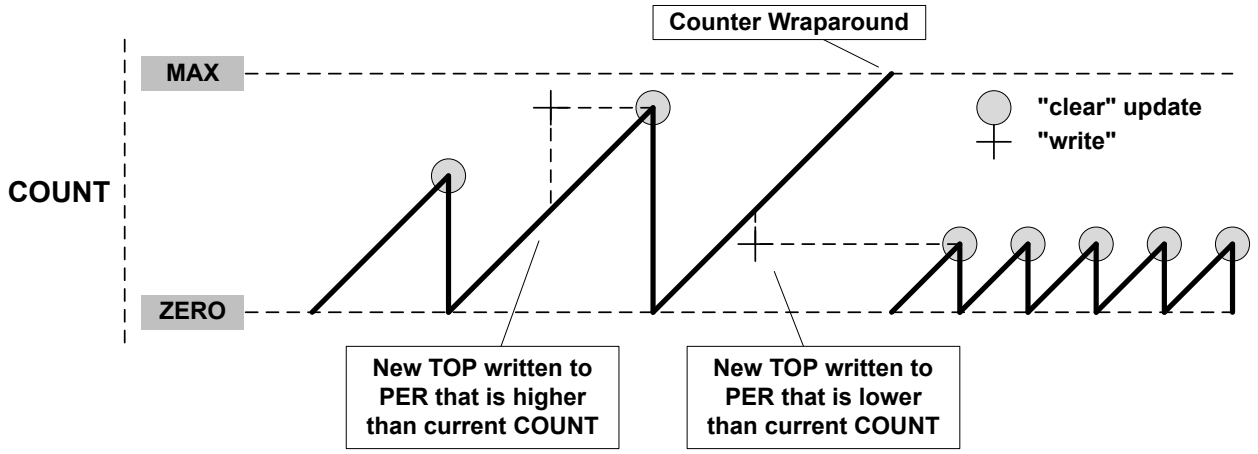
**Note:** In NFRQ, MFRQ or PWM down-counting counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continously copied into the PER independently of update conditions.

### Changing the Period

The counter period can be changed by writing a new TOP value to the Period register (PER or CC0, depending on the waveform generation mode), which is available in 8-bit mode. Any period update on registers (PER or CCx) is effective after the synchronization delay.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

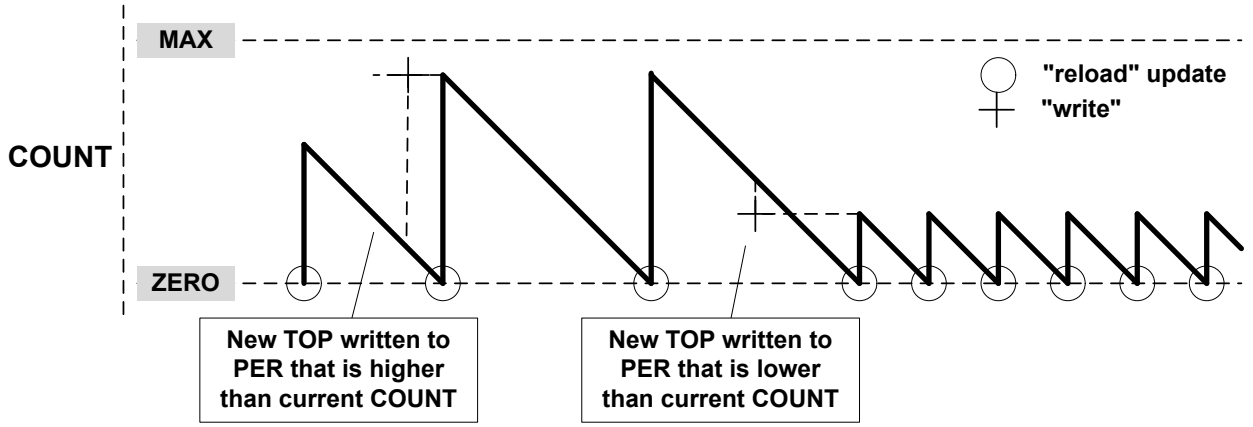
**Figure 48-8.** Unbuffered Single-Slope Up-Counting Operation




A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 48-8.

COUNT and TOP are continuously compared, so when a new TOP value that is lower than current COUNT is written to TOP, COUNT will wrap before a compare match.

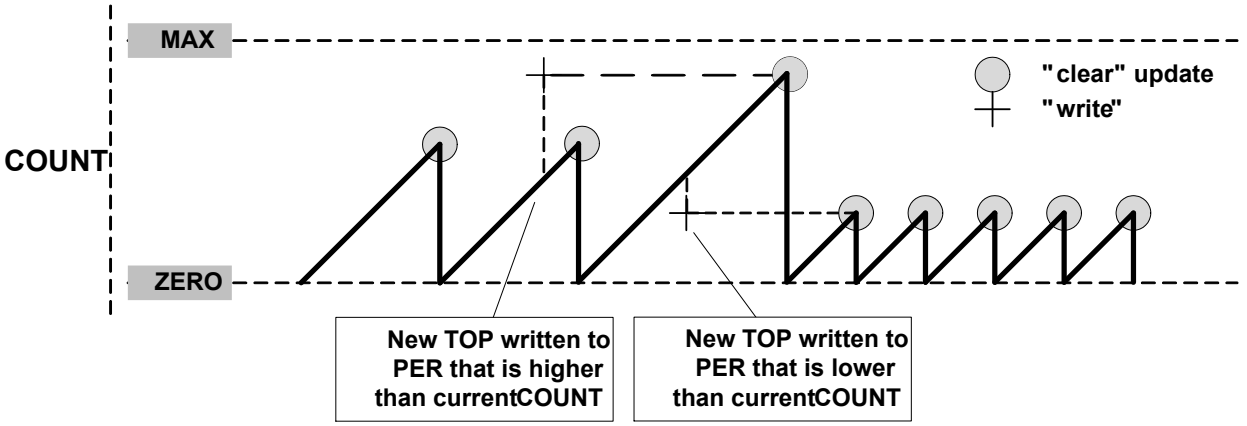
**Figure 48-9.** Unbuffered Single-Slope Down-Counting Operation




When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 48-10. This prevents wraparound and the generation of odd waveforms.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 48-10.** Changing the Period Using Buffering




### 48.6.2.8. Capture Operations

To enable and use capture operations, the corresponding Capture Channel x Enable bit in the Control A register (CTRLA.CAPTENx) must be written to '1'.

A capture trigger can be provided by input event line TC_EV or by asynchronous I/O pin WO[x] for each capture channel or by a TC event. To enable the capture from input event line, the Event Input Enable bit in the Event Control register (EVCTRL.TCEI) must be written to '1'. To enable the capture from the I/O pin, the Capture On Pin x Enable bit in the CTRLA register (CTRLA.COPENx) must be written to '1'.

**Notes:**

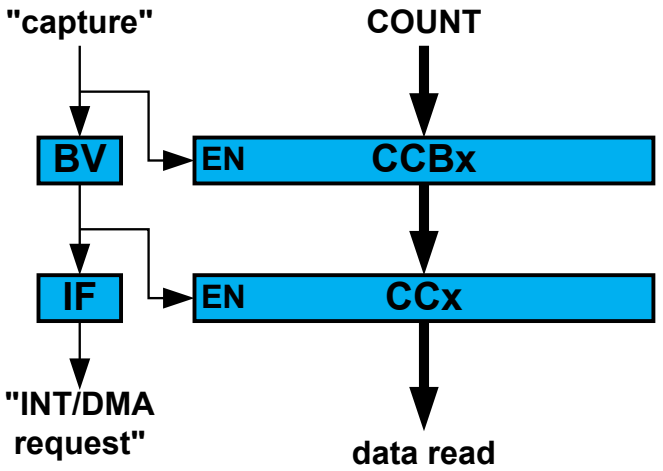
1. Capture on I/Os is only possible in 'Event' and 'Time-Stamp' capture action modes. Other modes can only use internal events. If I/Os toggling is needed in other modes, then the I/Os edge must be configured for generating internal events.
2. Capture on an event from the Event System is possible in 'Event', 'PPW/PWP/PW', and 'Time-Stamp' capture action modes. In this case, the event system channels must be configured to operate in asynchronous mode of operation.
3. Depending on CTRLA.COPENx, channel x can be configured for I/Os or internal event capture (both are mutually exclusive). One channel can be configured for I/Os capture while the other uses internal event capture.
4. Capture of the period and duty cycle on I/Os using PPW/PWP mode is possible using EIC and Event System.

By default, a capture operation is done when a rising edge is detected on the input signal. Capture on falling edge is available, its activation is depending on the input source:

- When the channel is used with a I/O pin, write a '1' to the corresponding Invert Enable bit in the Drive Control register (DRVCTRL.INVENx).
- When the channel is counting events from the Event System, write a '1' to the TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 48-11.** Capture Double Buffering



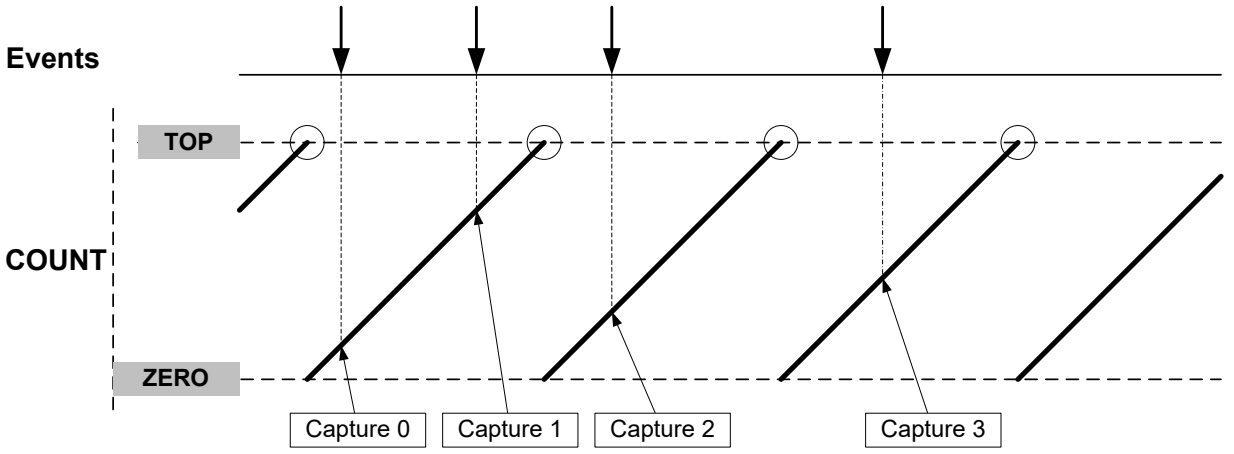

For input capture, the buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The buffer valid flag is passed to set the CCx interrupt flag (IF) and generate the optional interrupt, event or DMA request. The CCBUFx register value can't be read, all captured data must be read from the CCx register.

**Note:**
When up-counting (CTRLBSET.DIR = 0), counter values lower than '1' cannot be captured. To capture the full range including value '0', the TC must be in down-counting mode (CTRLBSET.DIR = 0).

#### 48.6.2.8.1. Event Capture Action on Events or I/Os

The compare and capture channels can be used as input capture channels to capture events from the Event System or I/O pins and give them a timestamp. This mode is selected when EVTCTRL.EVACT is configured either as OFF, RETRIGGER, COUNT or START. The following figure shows four capture events for one capture channel.

**Figure 48-12.** Input Capture Timing




The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

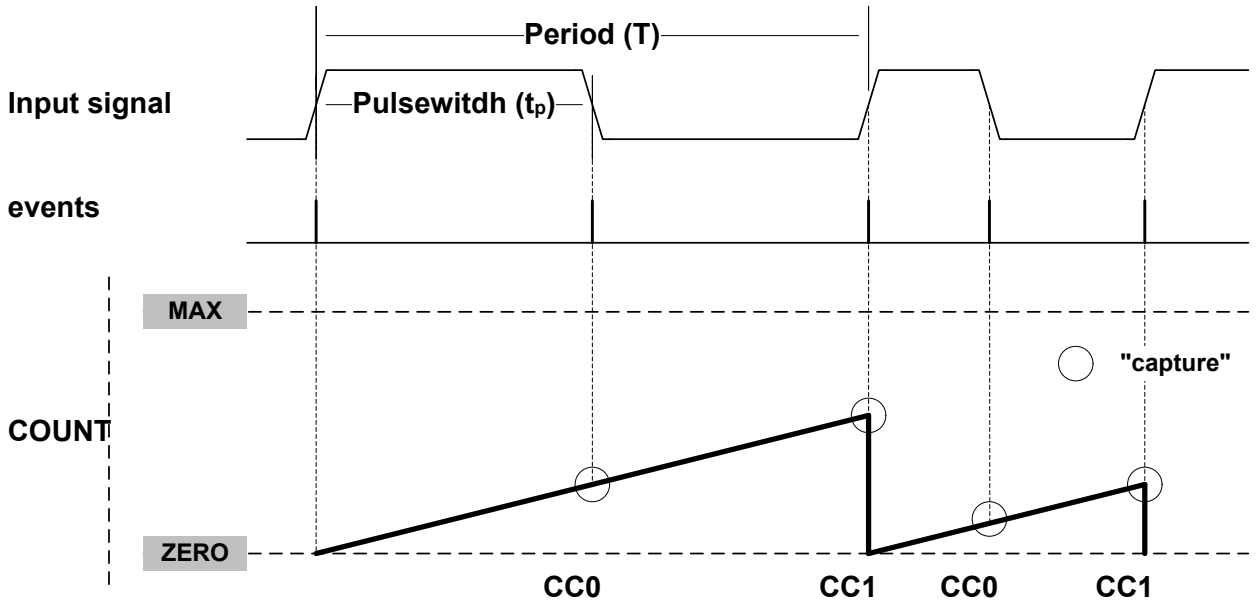
#### 48.6.2.8.2. Period and Pulse-Width (PPW/PWP) Capture Action on Events

The TC can perform two input captures and restart the counter on one of the edges. This enables the TC to measure the pulse width and period and to characterize the frequency *f* and duty cycle of an input signal:

$$f = \frac{1}{T}$$

$$\text{dutyCycle} = \frac{t_p}{T}$$

**Figure 48-13.** PWP Capture




Selecting PWP in the Event Action bit group in the Event Control register (EVCTRL.EVACT) enables the TC to perform one capture action on the rising edge and another one on the falling edge. The period *T* will be captured into CC1 and the pulse width $t_p$ in CC0. EVCTRL.EVACT = PPW (period and pulse-width) offers identical functionality, but will capture *T* into CC0 and $t_p$ into CC1.

The TC Event Input Invert Enable bit in the Event Control register (EVCTRL.TCINV) is used to select whether the wraparound must occur on the rising edge or the falling edge. If EVCTRL.TCINV = 1, the wraparound will happen on the falling edge.

The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

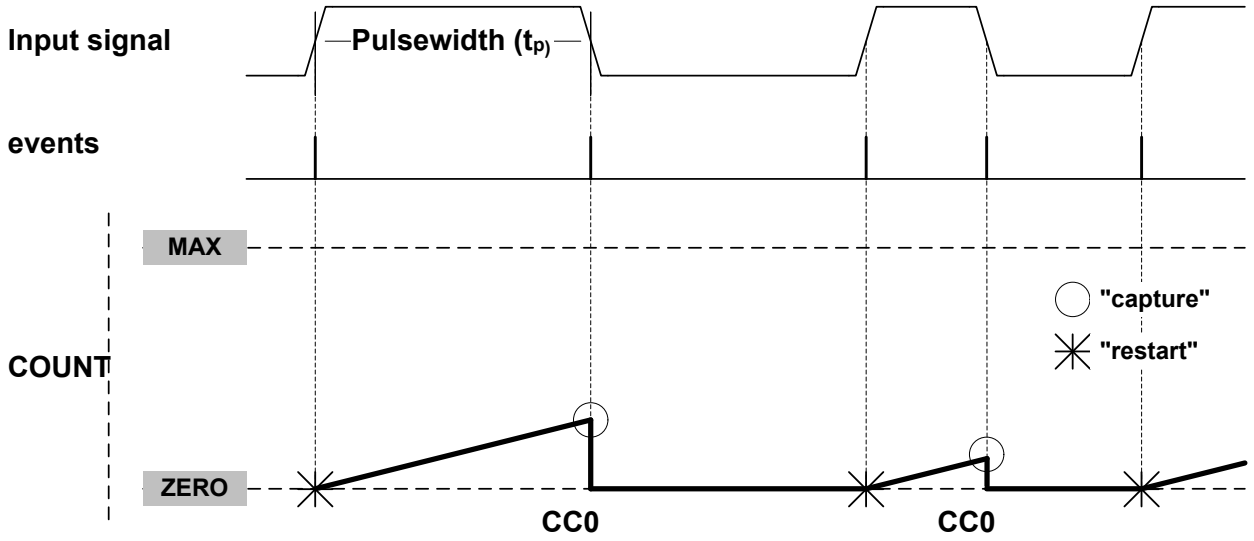
**Note:** The corresponding capture is working only if the channel is enabled in capture mode (CTRLA.CAPTENx = 1). Consequently, both channels must be enabled to fully characterize the input.

#### 48.6.2.8.3. Pulse-Width (PW) Capture Action on Events

The TC performs the input capture on the falling edge of the input signal. When the edge is detected, the counter value is cleared and the TC stops counting. When a rising edge is detected on the input signal, the counter restarts the counting operation. To enable the operation on opposite edges, the input signal to capture must be inverted (refer to EVCTRL.TCEINV).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 48-14.** Pulse-Width Capture on Channel 0




The TC can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Interrupt flag (INTFLAG.MCx) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

## 48.6.3. Additional Features

### 48.6.3.1. One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next Counter Overflow or Underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is automatically set and the waveform outputs are set to zero.

One-shot operation is enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT), and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TC will count until an overflow or underflow occurs and stops counting operation. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event, or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.

### 48.6.3.2. Time-Stamp Capture on Events or I/Os

This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

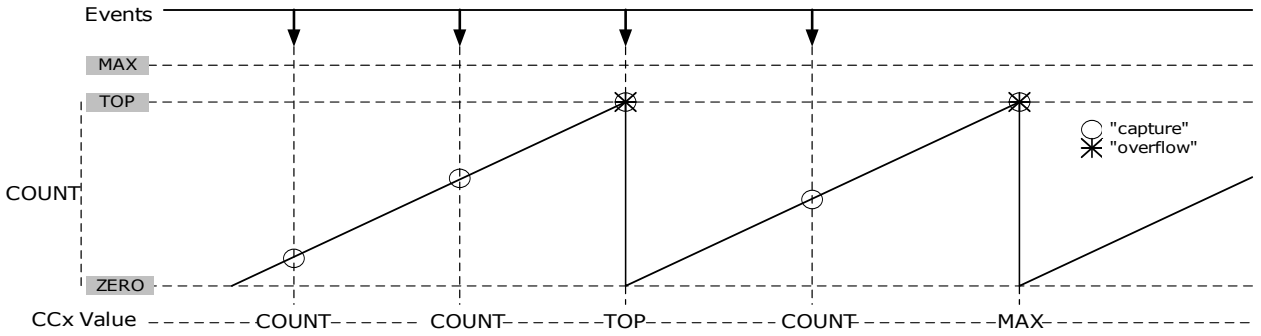
When a capture event from the Event System or the I/O pin is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 48-15.** Time Stamp




### 48.6.3.3. Minimum Capture

The minimum capture is enabled by writing the CAPTMIN mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMIN).

*CCx Content:*

In CAPTMIN operations, CCx keeps the Minimum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from zero. If the CCx register initial value is zero, no captures will be performed using the corresponding channel.

*MCx Behaviour:*

In CAPTMIN operation, capture is performed only when on capture event time, the counter value is lower than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is upper or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum value has been detected.

### 48.6.3.4. Maximum Capture

The maximum capture is enabled by writing the CAPTMAX mode in the Channel n Capture Mode bits in the Control A register (CTRLA.CAPTMODEn = CAPTMAX).

*CCx Content:*

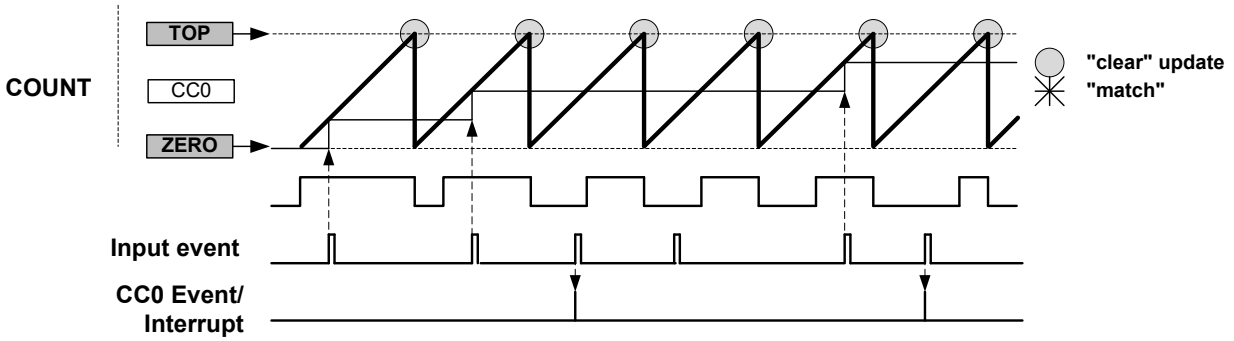
In CAPTMAX operations, CCx keeps the Maximum captured values. Before enabling this mode of capture, the user must initialize the corresponding CCx register value to a value different from TOP. If the CCx register initial value is TOP, no captures will be performed using the corresponding channel.

*MCx Behaviour:*

In CAPTMAX operation, capture is performed only when on capture event time, the counter value is upper than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is lower or equal to the value captured on the previous event. So interrupt flag is set when a new absolute local Maximum value has been detected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 48-16.** Maximum Capture Operation with CC0 Initialized with ZERO Value




### 48.6.4. DMA Operation

The TC can generate the following DMA requests:

- Overflow (OVF): the request is set when an update condition (overflow, underflow or re-trigger) is detected, the request is cleared by hardware on DMA acknowledge.
- Match or Capture Channel x (MCx): for a compare channel, the request is set on each compare match detection, the request is cleared by hardware on DMA acknowledge. For a capture channel, the request is set when valid data is present in the CCx register, and cleared when CCx register is read.

### 48.6.5. Interrupts

The TC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)
- Capture Overflow Error (ERR)

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs.

Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until either the interrupt flag is cleared, the interrupt is disabled, or the TC is reset. See INTFLAG for details on how to clear interrupt flags.

The TC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which interrupt condition is present.

Note that interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller

### 48.6.6. Events

The TC can generate the following output events:

- Overflow/Underflow (OVF)
- Match or Capture Channel x (MCx)

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEOx) enables the corresponding output event. The output event is disabled by writing EVCTRL.MCEOx=0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

One of the following event actions can be selected by the Event Action bit group in the Event Control register (EVCTRL.EVACT):

- Disable event action (OFF)
- Start TC (START)
- Re-trigger TC (RETRIGGER)
- Count on event (COUNT)
- Capture time stamp (STAMP)
- Capture Period (PPW and PWP)
- Capture Pulse Width (PW)

Writing a '1' to the TC Event Input bit in the Event Control register (EVCTRL.TCEI) enables input events to the TC. Writing a '0' to this bit disables input events to the TC. The TC requires only asynchronous event inputs. For further details on how configuring the asynchronous events, refer to *EVSYS - Event System*.

**Related Links**
Event System (EVSYS)

### 48.6.7. Sleep Mode Operation

The TC can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. This peripheral can wake up the device from any sleep mode using interrupts or perform actions through the Event System.

If the On Demand bit in the Control A register (CTRLA.ONDEMAND) is written to '1', the module stops requesting its peripheral clock when the STOP bit in STATUS register (STATUS.STOP) is set to '1'. When a re-trigger or start condition is detected, the TC requests the clock before the operation starts.

### 48.6.8. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)
- Capture Channel Buffer Valid bit in STATUS register (STATUS.CCBUFVx)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Channel x Compare/Capture Value and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD).

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48.7. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1. Register Summary - 8-bit Mode

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ONDEMAND | RUNSTDBY | PRESCSYNC[1:0] | | MODE[1:0] | | ENABLE | SWRST |
| | | 15:8 | | | | | ALOCK | PRESCALER[2:0] | | |
| | | 23:16 | | | COPEN1 | COPEN0 | | | CAPTEN1 | CAPTEN0 |
| | | 31:24 | | | | CAPTMODE1[1:0] | | | CAPTMODE0[1:0] | |
| 0x04 | CTRLBCLR | 7:0 | | CMD[2:0] | | | | ONESHOT | LUPD | DIR |
| 0x05 | CTRLBSET | 7:0 | | CMD[2:0] | | | | ONESHOT | LUPD | DIR |
| 0x06 | EVCTRL | 7:0 | | | TCEI | TCINV | | EVACT[2:0] | | |
| | | 15:8 | | | MCEO1 | MCEO0 | | | | OVFEO |
| 0x08 | INTENCLR | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x09 | INTENSET | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x0A | INTFLAG | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x0B | STATUS | 7:0 | | | CCBUFV1 | CCBUFV0 | PERBUFV | | SLAVE | STOP |
| 0x0C | WAVE | 7:0 | | | | | | | WAVEGEN[1:0] | |
| 0x0D | DRVCTRL | 7:0 | | | | | | | INVEN1 | INVEN0 |
| 0x0E | Reserved | | | | | | | | | |
| 0x0F | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x10 | SYNCBUSY | 7:0 | CC1 | CC0 | | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| 0x15 ... 0x1A | Reserved | | | | | | | | | |
| 0x1B | PER | 7:0 | PER[7:0] | | | | | | | |
| 0x1C | CC0 | 7:0 | CC[7:0] | | | | | | | |
| 0x1D | CC1 | 7:0 | CC[7:0] | | | | | | | |
| 0x1E ... 0x2E | Reserved | | | | | | | | | |
| 0x2F | PERBUF | 7:0 | PERBUF[7:0] | | | | | | | |
| 0x30 | CCBUF0 | 7:0 | CCBUF[7:0] | | | | | | | |
| 0x31 | CCBUF1 | 7:0 | CCBUF[7:0] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | CAPTMODE1[1:0] | | | CAPTMODE0[1:0] | |
| Access | | | | R/W | R/W | | R/W | R/W |
| Reset | | | | 0 | 0 | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | COPEN1 | COPEN0 | | | CAPTEN1 | CAPTEN0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ALOCK | PRESCALER[2:0] | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | PRESCSYNC[1:0] | | MODE[1:0] | | ENABLE | SWRST |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 28:27 – CAPTMODE1[1:0]** Capture mode Channel 1
These bits select the channel 1 capture mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Default capture |
| 0x1 | CAPTMIN | Minimum capture |
| 0x2 | CAPTMAX | Maximum capture |
| 0x3 | - | Reserved |

**Bits 25:24 – CAPTMODE0[1:0]** Capture mode Channel 0
These bits select the channel 0 capture mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Default capture |
| 0x1 | CAPTMIN | Minimum capture |
| 0x2 | CAPTMAX | Maximum capture |
| 0x3 | - | Reserved |

**Bits 20, 21 – COPENx** Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

| Value | Description |
|---|---|
| 0 | Event from Event System is selected as trigger source for capture operation on channel x. |
| 1 | I/O pin is selected as trigger source for capture operation on channel x. |

**Bits 16, 17 – CAPTENx** Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.
These bits are not synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | CAPTEN disables capture on channel x. |
| 1 | CAPTEN enables capture on channel x. |

**Bit 11 – ALOCK** Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The LUPD bit is not affected on overflow/underflow, and re-trigger event. |
| 1 | The LUPD bit is set on each overflow/underflow or re-trigger event. |

**Bits 10:8 – PRESCALER[2:0]** Prescaler

These bits select the counter prescaler factor.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV1 | Prescaler: GCLK_TCx |
| 0x1 | DIV2 | Prescaler: GCLK_TCx/2 |
| 0x2 | DIV4 | Prescaler: GCLK_TCx/4 |
| 0x3 | DIV8 | Prescaler: GCLK_TCx/8 |
| 0x4 | DIV16 | Prescaler: GCLK_TCx/16 |
| 0x5 | DIV64 | Prescaler: GCLK_TCx/64 |
| 0x6 | DIV256 | Prescaler: GCLK_TCx/256 |
| 0x7 | DIV1024 | Prescaler: GCLK_TCx/1024 |

**Bit 7 – ONDEMAND** Clock On Demand

This bit selects the clock requirements when the TC is stopped.
In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1). |
| 1 | The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected. |

**Bit 6 – RUNSTDBY** Run in Standby

This bit is used to keep the TC running in standby mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The TC is halted in standby. |
| 1 | The TC continues to run in standby. |

**Bits 5:4 – PRESCSYNC[1:0]** Prescaler and Counter Synchronization

These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | GCLK | Reload or reset the counter on next generic clock |
| 0x1 | PRESC | Reload or reset the counter on next prescaler clock |
| 0x2 | RESYNC | Reload or reset the counter on next generic clock. Reset the prescaler counter |
| 0x3 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 3:2 – MODE[1:0]** Timer Counter Mode
These bits select the counter mode.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | COUNT16 | Counter in 16-bit mode |
| 0x1 | COUNT8 | Counter in 8-bit mode |
| 0x2 | COUNT32 | Counter in 32-bit mode |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE** Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.2. Control B Clear

**Name:** CTRLBCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** Command
Writing a `0' to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

**Bit 2 – ONESHOT** One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable one-shot operation.

| Value | Description |
|---|---|
| 0 | The TC will wrap around and continue counting on an overflow/underflow condition. |
| 1 | The TC will wrap around and stop on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the LUPD bit.

| Value | Description |
|---|---|
| 0 | The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.3. Control B Set

**Name:** CTRLBSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TCx clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 to this bit field will issue a command for execution.

**Important:** This command requires synchronization before being executed.
A valid sequence is:the following:

- Issue CMD command (CTRLBSET.CMD = command)
- Wait for CMD synchronization (SYNCBUSY.CTRLB = 0)
- Wait for CMD read back as zero (CTRLBSET.CMD = 0)

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | RETRIGGER | Force a start, restart or retrigger |
| 0x2 | STOP | Force a stop |
| 0x3 | UPDATE | Force update of double buffered registers |
| 0x4 | READSYNC | Force a read synchronization of COUNT |

**Bit 2 – ONESHOT** One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable one-shot operation.

| Value | Description |
|---|---|
| 0 | The TC will wrap around and continue counting on an overflow/underflow condition. |
| 1 | The TC will wrap around and stop on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the LUPD bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit has no effect when input capture operation is enabled.

| Value | Description |
|---|---|
| 0 | The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction

This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.4. Event Control

**Name:** EVCTRL
**Offset:** 0x06
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | MCEO1 | MCEO0 | | | | OVFEO |
| Access | | | R/W | R/W | | | | R/W |
| Reset | | | 0 | 0 | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | TCEI | TCINV | | EVACT[2:0] | | |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 0 | 0 | | 0 | 0 | 0 |

**Bit 13 – MCEO1** Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

| Value | Description |
|---|---|
| 0 | Match/Capture event on channel x is disabled and will not be generated. |
| 1 | Match/Capture event on channel x is enabled and will be generated for every compare/capture. |

**Bit 12 – MCEO0** Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

| Value | Description |
|---|---|
| 0 | Match/Capture event on channel x is disabled and will not be generated. |
| 1 | Match/Capture event on channel x is enabled and will be generated for every compare/capture. |

**Bit 8 – OVFEO** Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

| Value | Description |
|---|---|
| 0 | Overflow/Underflow event is disabled and will not be generated. |
| 1 | Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow. |

**Bit 5 – TCEI** TC Event Enable
This bit is used to enable asynchronous input events to the TC.

| Value | Description |
|---|---|
| 0 | Incoming events are disabled. |
| 1 | Incoming events are enabled. |

**Bit 4 – TCINV** TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

| Value | Description |
|---|---|
| 0 | Input event source is not inverted. |
| 1 | Input event source is inverted. |

**Bits 2:0 – EVACT[2:0]** Event Action
These bits define the event action the TC will perform on an event.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Event action disabled |
| 0x1 | RETRIGGER | Start, restart or retrigger TC on event |
| 0x2 | COUNT | Count on event |
| 0x3 | START | Start TC on event |
| 0x4 | STAMP | Time stamp capture |
| 0x5 | PPW | Period captured in CC0, pulse width in CC1 |
| 0x6 | PWP | Period captured in CC1, pulse width in CC0 |
| 0x7 | PW | Pulse width capture |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.1.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 4 – MC0** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 4 – MC0** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

**Bit 4 – MC0** Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

**Bit 1 – ERR** Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

**Bit 0 – OVF** Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.8. Status

**Name:** STATUS
**Offset:** 0x0B
**Reset:** 0x01
**Property:** Read-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | CCBUFV1 | CCBUFV0 | PERBUFV | | SLAVE | STOP |
| Access | | | R/W | R/W | R/W | | R | R |
| Reset | | | 0 | 0 | 0 | | 0 | 1 |

**Bits 4, 5 – CCBUFV** Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.
The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

**Bit 3 – PERBUFV** Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

**Bit 1 – SLAVE** Client Status Flag
This bit is only available in 32-bit mode on the Client TC (i.e., TC1 and/or TC3). The bit is set when the associated Host TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

**Bit 0 – STOP** Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

| Value | Description |
|---|---|
| 0 | Counter is running. |
| 1 | Counter is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.9. Waveform Generation Control

**Name:** WAVE
**Offset:** 0x0C
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | WAVEGEN[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – WAVEGEN[1:0]** Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in Waveform Output Operations.
These bits are not synchronized.

| Value | Name | Operation | Top Value | Output Waveform on Match | Output Waveform on Wraparound |
|---|---|---|---|---|---|
| 0x0 | NFRQ | Normal frequency | PER[1] / Max | Toggle | No action |
| 0x1 | MFRQ | Match frequency | CC0 | Toggle | No action |
| 0x2 | NPWM | Normal PWM | PER[1] / Max | Set | Clear |
| 0x3 | MPWM | Match PWM | CC0 | Set | Clear |

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.1.10. Driver Control

**Name:** DRVCTRL
**Offset:** 0x0D
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | INVEN1 | INVEN0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 0, 1 – INVENx** Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

| Value | Description |
|---|---|
| 0 | Disable inversion of the WO[x] output and IO input pin. |
| 1 | Enable inversion of the WO[x] output and IO input pin. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.11. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0F
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

| Value | Description |
|---|---|
| 0 | The TC is halted when the device is halted in debug mode. |
| 1 | The TC continues normal operation when the device is halted in debug mode. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.1.12. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC1 | CC0 | | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| Access | R | R | | R | R | R | R | R |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 0 |

**Bits 6, 7 – CCx** Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

**Bit 4 – COUNT** COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

**Bit 3 – STATUS** STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

**Bit 2 – CTRLB** CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

**Bit 1 – ENABLE** ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

**Bit 0 – SWRST** SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.13. Counter Value, 8-bit Mode

**Name:** COUNT
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

**Note:** Prior to any read access, this register must be synchronized by the user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD = READSYNC).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – COUNT[7:0]** Counter Value
These bits contain the current counter value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.14. Period Value, 8-bit Mode

**Name:** PER
**Offset:** 0x1B
**Reset:** 0xFF
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 7:0 – PER[7:0]** Period Value
These bits hold the value of the TC period count.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.15. Channel x Compare/Capture Value, 8-bit Mode

**Name:** CCx
**Offset:** 0x1C + x*0x01 [x=0..1]
**Reset:** 0x00
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – CC[7:0]** Channel x Compare/Capture Value
These bits contain the compare/capture value in 8-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.16. Period Buffer Value, 8-bit Mode

**Name:** PERBUF
**Offset:** 0x2F
**Reset:** 0xFF
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PERBUF[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 7:0 – PERBUF[7:0]** Period Buffer Value
These bits hold the value of the period buffer register. The value is copied to PER register on UPDATE condition.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.1.17. Channel x Compare Buffer Value, 8-bit Mode

**Name:** CCBUFx
**Offset:** 0x30 + x*0x01 [x=0..1]
**Reset:** 0x00
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – CCBUF[7:0]** Channel x Compare Buffer Value
These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2. Register Summary - 16-bit Mode

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ONDEMAND | RUNSTDBY | PRESCSYNC[1:0] | | MODE[1:0] | | ENABLE | SWRST |
| | | 15:8 | | | | | ALOCK | PRESCALER[2:0] | | |
| | | 23:16 | | | COPEN1 | COPEN0 | | | CAPTEN1 | CAPTEN0 |
| | | 31:24 | | | | CAPTMODE1[1:0] | | | CAPTMODE0[1:0] | |
| 0x04 | CTRLBCLR | 7:0 | | CMD[2:0] | | | | ONESHOT | LUPD | DIR |
| 0x05 | CTRLBSET | 7:0 | | CMD[2:0] | | | | ONESHOT | LUPD | DIR |
| 0x06 | EVCTRL | 7:0 | | | TCEI | TCINV | | EVACT[2:0] | | |
| | | 15:8 | | | MCEO1 | MCEO0 | | | | OVFEO |
| 0x08 | INTENCLR | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x09 | INTENSET | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x0A | INTFLAG | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x0B | STATUS | 7:0 | | | CCBUFV1 | CCBUFV0 | PERBUFV | | SLAVE | STOP |
| 0x0C | WAVE | 7:0 | | | | | | | WAVEGEN[1:0] | |
| 0x0D | DRVCTRL | 7:0 | | | | | | | INVEN1 | INVEN0 |
| 0x0E | Reserved | | | | | | | | | |
| 0x0F | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x10 | SYNCBUSY | 7:0 | CC1 | CC0 | | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| 0x16 ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | CC0 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| 0x1E | CC1 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| 0x20 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | CCBUF0 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |
| 0x32 | CCBUF1 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | CAPTMODE1[1:0] | | | CAPTMODE0[1:0] | |
| Access | | | | R/W | R/W | | R/W | R/W |
| Reset | | | | 0 | 0 | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | COPEN1 | COPEN0 | | | CAPTEN1 | CAPTEN0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ALOCK | PRESCALER[2:0] | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | PRESCSYNC[1:0] | | MODE[1:0] | | ENABLE | SWRST |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 28:27 – CAPTMODE1[1:0]** Capture mode Channel 1
These bits select the channel 1 capture mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Default capture |
| 0x1 | CAPTMIN | Minimum capture |
| 0x2 | CAPTMAX | Maximum capture |
| 0x3 | - | Reserved |

**Bits 25:24 – CAPTMODE0[1:0]** Capture mode Channel 0
These bits select the channel 0 capture mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Default capture |
| 0x1 | CAPTMIN | Minimum capture |
| 0x2 | CAPTMAX | Maximum capture |
| 0x3 | - | Reserved |

**Bits 20, 21 – COPENx** Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

| Value | Description |
|---|---|
| 0 | Event from Event System is selected as trigger source for capture operation on channel x. |
| 1 | I/O pin is selected as trigger source for capture operation on channel x. |

**Bits 16, 17 – CAPTENx** Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.
These bits are not synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | CAPTEN disables capture on channel x. |
| 1 | CAPTEN enables capture on channel x. |

**Bit 11 – ALOCK** Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The LUPD bit is not affected on overflow/underflow, and re-trigger event. |
| 1 | The LUPD bit is set on each overflow/underflow or re-trigger event. |

**Bits 10:8 – PRESCALER[2:0]** Prescaler

These bits select the counter prescaler factor.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV1 | Prescaler: GCLK_TCx |
| 0x1 | DIV2 | Prescaler: GCLK_TCx/2 |
| 0x2 | DIV4 | Prescaler: GCLK_TCx/4 |
| 0x3 | DIV8 | Prescaler: GCLK_TCx/8 |
| 0x4 | DIV16 | Prescaler: GCLK_TCx/16 |
| 0x5 | DIV64 | Prescaler: GCLK_TCx/64 |
| 0x6 | DIV256 | Prescaler: GCLK_TCx/256 |
| 0x7 | DIV1024 | Prescaler: GCLK_TCx/1024 |

**Bit 7 – ONDEMAND** Clock On Demand

This bit selects the clock requirements when the TC is stopped.
In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1). |
| 1 | The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected. |

**Bit 6 – RUNSTDBY** Run in Standby

This bit is used to keep the TC running in standby mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The TC is halted in standby. |
| 1 | The TC continues to run in standby. |

**Bits 5:4 – PRESCSYNC[1:0]** Prescaler and Counter Synchronization

These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | GCLK | Reload or reset the counter on next generic clock |
| 0x1 | PRESC | Reload or reset the counter on next prescaler clock |
| 0x2 | RESYNC | Reload or reset the counter on next generic clock. Reset the prescaler counter |
| 0x3 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 3:2 – MODE[1:0]** Timer Counter Mode
These bits select the counter mode.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | COUNT16 | Counter in 16-bit mode |
| 0x1 | COUNT8 | Counter in 8-bit mode |
| 0x2 | COUNT32 | Counter in 32-bit mode |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE** Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.2. Control B Clear

**Name:** CTRLBCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** Command
Writing a `0' to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

**Bit 2 – ONESHOT** One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable one-shot operation.

| Value | Description |
|---|---|
| 0 | The TC will wrap around and continue counting on an overflow/underflow condition. |
| 1 | The TC will wrap around and stop on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the LUPD bit.

| Value | Description |
|---|---|
| 0 | The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.3. Control B Set

**Name:** CTRLBSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TCx clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 to this bit field will issue a command for execution.

**Important:** This command requires synchronization before being executed.
A valid sequence is:the following:

- Issue CMD command (CTRLBSET.CMD = command)
- Wait for CMD synchronization (SYNCBUSY.CTRLB = 0)
- Wait for CMD read back as zero (CTRLBSET.CMD = 0)

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | RETRIGGER | Force a start, restart or retrigger |
| 0x2 | STOP | Force a stop |
| 0x3 | UPDATE | Force update of double buffered registers |
| 0x4 | READSYNC | Force a read synchronization of COUNT |

**Bit 2 – ONESHOT** One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable one-shot operation.

| Value | Description |
|---|---|
| 0 | The TC will wrap around and continue counting on an overflow/underflow condition. |
| 1 | The TC will wrap around and stop on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the LUPD bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit has no effect when input capture operation is enabled.

| Value | Description |
|---|---|
| 0 | The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction

This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.4. Event Control

**Name:** EVCTRL
**Offset:** 0x06
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | MCEO1 | MCEO0 | | | | OVFEO |
| Access | | | R/W | R/W | | | | R/W |
| Reset | | | 0 | 0 | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | TCEI | TCINV | | EVACT[2:0] | | |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 0 | 0 | | 0 | 0 | 0 |

**Bit 13 – MCEO1** Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

| Value | Description |
|---|---|
| 0 | Match/Capture event on channel x is disabled and will not be generated. |
| 1 | Match/Capture event on channel x is enabled and will be generated for every compare/capture. |

**Bit 12 – MCEO0** Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

| Value | Description |
|---|---|
| 0 | Match/Capture event on channel x is disabled and will not be generated. |
| 1 | Match/Capture event on channel x is enabled and will be generated for every compare/capture. |

**Bit 8 – OVFEO** Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

| Value | Description |
|---|---|
| 0 | Overflow/Underflow event is disabled and will not be generated. |
| 1 | Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow. |

**Bit 5 – TCEI** TC Event Enable
This bit is used to enable asynchronous input events to the TC.

| Value | Description |
|---|---|
| 0 | Incoming events are disabled. |
| 1 | Incoming events are enabled. |

**Bit 4 – TCINV** TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

| Value | Description |
|---|---|
| 0 | Input event source is not inverted. |
| 1 | Input event source is inverted. |

**Bits 2:0 – EVACT[2:0]** Event Action
These bits define the event action the TC will perform on an event.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Event action disabled |
| 0x1 | RETRIGGER | Start, restart or retrigger TC on event |
| 0x2 | COUNT | Count on event |
| 0x3 | START | Start TC on event |
| 0x4 | STAMP | Time stamp capture |
| 0x5 | PPW | Period captured in CC0, pulse width in CC1 |
| 0x6 | PWP | Period captured in CC1, pulse width in CC0 |
| 0x7 | PW | Pulse width capture |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.2.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 4 – MC0** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 4 – MC0** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

**Bit 4 – MC0** Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

**Bit 1 – ERR** Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

**Bit 0 – OVF** Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.8. Status

**Name:** STATUS
**Offset:** 0x0B
**Reset:** 0x01
**Property:** Read-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | CCBUFV1 | CCBUFV0 | PERBUFV | | SLAVE | STOP |
| Access | | | R/W | R/W | R/W | | R | R |
| Reset | | | 0 | 0 | 0 | | 0 | 1 |

**Bits 4, 5 – CCBUFV** Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.
The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

**Bit 3 – PERBUFV** Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

**Bit 1 – SLAVE** Client Status Flag
This bit is only available in 32-bit mode on the Client TC (i.e., TC1 and/or TC3). The bit is set when the associated Host TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

**Bit 0 – STOP** Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

| Value | Description |
|---|---|
| 0 | Counter is running. |
| 1 | Counter is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.9. Waveform Generation Control

**Name:** WAVE
**Offset:** 0x0C
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | WAVEGEN[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – WAVEGEN[1:0]** Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in Waveform Output Operations.
These bits are not synchronized.

| Value | Name | Operation | Top Value | Output Waveform on Match | Output Waveform on Wraparound |
|---|---|---|---|---|---|
| 0x0 | NFRQ | Normal frequency | PER[1] / Max | Toggle | No action |
| 0x1 | MFRQ | Match frequency | CC0 | Toggle | No action |
| 0x2 | NPWM | Normal PWM | PER[1] / Max | Set | Clear |
| 0x3 | MPWM | Match PWM | CC0 | Set | Clear |

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.2.10. Driver Control

**Name:** DRVCTRL
**Offset:** 0x0D
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | INVEN1 | INVEN0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 0, 1 – INVENx** Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

| Value | Description |
|---|---|
| 0 | Disable inversion of the WO[x] output and IO input pin. |
| 1 | Enable inversion of the WO[x] output and IO input pin. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.11. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0F
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

| Value | Description |
|---|---|
| 0 | The TC is halted when the device is halted in debug mode. |
| 1 | The TC continues normal operation when the device is halted in debug mode. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.2.12. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC1 | CC0 | | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| Access | R | R | | R | R | R | R | R |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 0 |

**Bits 6, 7 – CCx** Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

**Bit 4 – COUNT** COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

**Bit 3 – STATUS** STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

**Bit 2 – CTRLB** CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

**Bit 1 – ENABLE** ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

**Bit 0 – SWRST** SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.13. Counter Value, 16-bit Mode

**Name:** COUNT
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

**Note:** Prior to any read access, this register must be synchronized by the user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD = READSYNC).

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – COUNT[15:0]** Counter Value
These bits contain the current counter value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.14. Channel x Compare/Capture Value, 16-bit Mode

**Name:** CCx
**Offset:** 0x1C + x*0x02 [x=0..1]
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – CC[15:0]** Channel x Compare/Capture Value
These bits contain the compare/capture value in 16-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.2.15. Channel x Compare Buffer Value, 16-bit Mode

**Name:** CCBUFx
**Offset:** 0x30 + x*0x02 [x=0..1]
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – CCBUF[15:0]** Channel x Compare Buffer Value

These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3. Register Summary - 32-bit Mode

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ONDEMAND | RUNSTDBY | PRESCSYNC[1:0] | | MODE[1:0] | | ENABLE | SWRST |
| | | 15:8 | | | | | ALOCK | PRESCALER[2:0] | | |
| | | 23:16 | | | COPEN1 | COPEN0 | | | CAPTEN1 | CAPTEN0 |
| | | 31:24 | | | | CAPTMODE1[1:0] | | | CAPTMODE0[1:0] | |
| 0x04 | CTRLBCLR | 7:0 | | CMD[2:0] | | | | ONESHOT | LUPD | DIR |
| 0x05 | CTRLBSET | 7:0 | | CMD[2:0] | | | | ONESHOT | LUPD | DIR |
| 0x06 | EVCTRL | 7:0 | | | TCEI | TCINV | | EVACT[2:0] | | |
| | | 15:8 | | | MCEO1 | MCEO0 | | | | OVFEO |
| 0x08 | INTENCLR | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x09 | INTENSET | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x0A | INTFLAG | 7:0 | | | MC1 | MC0 | | | ERR | OVF |
| 0x0B | STATUS | 7:0 | | | CCBUFV1 | CCBUFV0 | PERBUFV | | SLAVE | STOP |
| 0x0C | WAVE | 7:0 | | | | | | | WAVEGEN[1:0] | |
| 0x0D | DRVCTRL | 7:0 | | | | | | | INVEN1 | INVEN0 |
| 0x0E | Reserved | | | | | | | | | |
| 0x0F | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x10 | SYNCBUSY | 7:0 | CC1 | CC0 | | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | COUNT[23:16] | | | | | | | |
| | | 31:24 | COUNT[31:24] | | | | | | | |
| 0x18 ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | CC0 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| | | 23:16 | CC[23:16] | | | | | | | |
| | | 31:24 | CC[31:24] | | | | | | | |
| 0x20 | CC1 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| | | 23:16 | CC[23:16] | | | | | | | |
| | | 31:24 | CC[31:24] | | | | | | | |
| 0x24 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | CCBUF0 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |
| | | 23:16 | CCBUF[23:16] | | | | | | | |
| | | 31:24 | CCBUF[31:24] | | | | | | | |
| 0x34 | CCBUF1 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |
| | | 23:16 | CCBUF[23:16] | | | | | | | |
| | | 31:24 | CCBUF[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | CAPTMODE1[1:0] | | | CAPTMODE0[1:0] | |
| Access | | | | R/W | R/W | | R/W | R/W |
| Reset | | | | 0 | 0 | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | COPEN1 | COPEN0 | | | CAPTEN1 | CAPTEN0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ALOCK | PRESCALER[2:0] | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | PRESCSYNC[1:0] | | MODE[1:0] | | ENABLE | SWRST |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 28:27 – CAPTMODE1[1:0]** Capture mode Channel 1
These bits select the channel 1 capture mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Default capture |
| 0x1 | CAPTMIN | Minimum capture |
| 0x2 | CAPTMAX | Maximum capture |
| 0x3 | - | Reserved |

**Bits 25:24 – CAPTMODE0[1:0]** Capture mode Channel 0
These bits select the channel 0 capture mode.

| Value | Name | Description |
|---|---|---|
| 0x0 | DEFAULT | Default capture |
| 0x1 | CAPTMIN | Minimum capture |
| 0x2 | CAPTMAX | Maximum capture |
| 0x3 | - | Reserved |

**Bits 20, 21 – COPENx** Capture On Pin x Enable
Bit x of COPEN[1:0] selects the trigger source for capture operation, either events or I/O pin input.

| Value | Description |
|---|---|
| 0 | Event from Event System is selected as trigger source for capture operation on channel x. |
| 1 | I/O pin is selected as trigger source for capture operation on channel x. |

**Bits 16, 17 – CAPTENx** Capture Channel x Enable
Bit x of CAPTEN[1:0] selects whether channel x is a capture or a compare channel.
These bits are not synchronized.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | CAPTEN disables capture on channel x. |
| 1 | CAPTEN enables capture on channel x. |

**Bit 11 – ALOCK** Auto Lock

When this bit is set, Lock bit update (LUPD) is set to '1' on each overflow/underflow or re-trigger event.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The LUPD bit is not affected on overflow/underflow, and re-trigger event. |
| 1 | The LUPD bit is set on each overflow/underflow or re-trigger event. |

**Bits 10:8 – PRESCALER[2:0]** Prescaler

These bits select the counter prescaler factor.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV1 | Prescaler: GCLK_TCx |
| 0x1 | DIV2 | Prescaler: GCLK_TCx/2 |
| 0x2 | DIV4 | Prescaler: GCLK_TCx/4 |
| 0x3 | DIV8 | Prescaler: GCLK_TCx/8 |
| 0x4 | DIV16 | Prescaler: GCLK_TCx/16 |
| 0x5 | DIV64 | Prescaler: GCLK_TCx/64 |
| 0x6 | DIV256 | Prescaler: GCLK_TCx/256 |
| 0x7 | DIV1024 | Prescaler: GCLK_TCx/1024 |

**Bit 7 – ONDEMAND** Clock On Demand

This bit selects the clock requirements when the TC is stopped.
In standby mode, if the Run in Standby bit (CTRLA.RUNSTDBY) is '0', ONDEMAND is forced to '0'.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The On Demand is disabled. If On Demand is disabled, the TC will continue to request the clock when its operation is stopped (STATUS.STOP=1). |
| 1 | The On Demand is enabled. When On Demand is enabled, the stopped TC will not request the clock. The clock is requested when a software re-trigger command is applied or when an event with start/re-trigger action is detected. |

**Bit 6 – RUNSTDBY** Run in Standby

This bit is used to keep the TC running in standby mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The TC is halted in standby. |
| 1 | The TC continues to run in standby. |

**Bits 5:4 – PRESCSYNC[1:0]** Prescaler and Counter Synchronization

These bits select whether the counter should wrap around on the next GCLK_TCx clock or the next prescaled GCLK_TCx clock. It also makes it possible to reset the prescaler.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | GCLK | Reload or reset the counter on next generic clock |
| 0x1 | PRESC | Reload or reset the counter on next prescaler clock |
| 0x2 | RESYNC | Reload or reset the counter on next generic clock. Reset the prescaler counter |
| 0x3 | - | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 3:2 – MODE[1:0]** Timer Counter Mode
These bits select the counter mode.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | COUNT16 | Counter in 16-bit mode |
| 0x1 | COUNT8 | Counter in 8-bit mode |
| 0x2 | COUNT32 | Counter in 32-bit mode |
| 0x3 | - | Reserved |

**Bit 1 – ENABLE** Enable
Due to synchronization, there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the ENABLE Synchronization Busy bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.
This bit is not enable protected.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TC, except DBGCTRL, to their initial state, and the TC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.2. Control B Clear

**Name:** CTRLBCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to clear bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set register (CTRLBSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** Command
Writing a `0' to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

**Bit 2 – ONESHOT** One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable one-shot operation.

| Value | Description |
|---|---|
| 0 | The TC will wrap around and continue counting on an overflow/underflow condition. |
| 1 | The TC will wrap around and stop on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the LUPD bit.

| Value | Description |
|---|---|
| 0 | The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction
This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.3. Control B Set

**Name:** CTRLBSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to set bits in the CTRLB register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear register (CTRLBCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** Command
These bits are used for software control of the TC. The commands are executed on the next prescaled GCLK_TCx clock cycle. When a command has been executed, the CMD bit group will be read back as zero.
Writing 0x0 to these bits has no effect.
Writing a value different from 0x0 to this bit field will issue a command for execution.

**Important:** This command requires synchronization before being executed.
A valid sequence is:the following:

- Issue CMD command (CTRLBSET.CMD = command)
- Wait for CMD synchronization (SYNCBUSY.CTRLB = 0)
- Wait for CMD read back as zero (CTRLBSET.CMD = 0)

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | RETRIGGER | Force a start, restart or retrigger |
| 0x2 | STOP | Force a stop |
| 0x3 | UPDATE | Force update of double buffered registers |
| 0x4 | READSYNC | Force a read synchronization of COUNT |

**Bit 2 – ONESHOT** One-Shot on Counter
This bit controls one-shot operation of the TC.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable one-shot operation.

| Value | Description |
|---|---|
| 0 | The TC will wrap around and continue counting on an overflow/underflow condition. |
| 1 | The TC will wrap around and stop on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the LUPD bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit has no effect when input capture operation is enabled.

| Value | Description |
|---|---|
| 0 | The CCBUFx and PERBUF buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The CCBUFx and PERBUF buffer registers value are not copied into CCx and PER registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction

This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.4. Event Control

**Name:** EVCTRL
**Offset:** 0x06
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | MCEO1 | MCEO0 | | | | OVFEO |
| Access | | | R/W | R/W | | | | R/W |
| Reset | | | 0 | 0 | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | TCEI | TCINV | | EVACT[2:0] | | |
| Access | | | R/W | R/W | | R/W | R/W | R/W |
| Reset | | | 0 | 0 | | 0 | 0 | 0 |

**Bit 13 – MCEO1** Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

| Value | Description |
|---|---|
| 0 | Match/Capture event on channel x is disabled and will not be generated. |
| 1 | Match/Capture event on channel x is enabled and will be generated for every compare/capture. |

**Bit 12 – MCEO0** Match or Capture Channel x Event Output Enable [x = 1..0]
These bits enable the generation of an event for every match or capture on channel x.

| Value | Description |
|---|---|
| 0 | Match/Capture event on channel x is disabled and will not be generated. |
| 1 | Match/Capture event on channel x is enabled and will be generated for every compare/capture. |

**Bit 8 – OVFEO** Overflow/Underflow Event Output Enable
This bit enables the Overflow/Underflow event. When enabled, an event will be generated when the counter overflows/underflows.

| Value | Description |
|---|---|
| 0 | Overflow/Underflow event is disabled and will not be generated. |
| 1 | Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow. |

**Bit 5 – TCEI** TC Event Enable
This bit is used to enable asynchronous input events to the TC.

| Value | Description |
|---|---|
| 0 | Incoming events are disabled. |
| 1 | Incoming events are enabled. |

**Bit 4 – TCINV** TC Inverted Event Input Polarity
This bit inverts the asynchronous input event source.

| Value | Description |
|---|---|
| 0 | Input event source is not inverted. |
| 1 | Input event source is inverted. |

**Bits 2:0 – EVACT[2:0]** Event Action
These bits define the event action the TC will perform on an event.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Event action disabled |
| 0x1 | RETRIGGER | Start, restart or retrigger TC on event |
| 0x2 | COUNT | Count on event |
| 0x3 | START | Start TC on event |
| 0x4 | STAMP | Time stamp capture |
| 0x5 | PPW | Period captured in CC0, pulse width in CC1 |
| 0x6 | PWP | Period captured in CC1, pulse width in CC0 |
| 0x7 | PW | Pulse width capture |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.3.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 4 – MC0** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will clear the corresponding Match or Capture Channel x Interrupt Enable bit, which disables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Enable bit, which disables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear register (INTENCLR).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 4 – MC0** Match or Capture Channel x Interrupt Enable
Writing a '0' to these bits has no effect.
Writing a '1' to MCx will set the corresponding Match or Capture Channel x Interrupt Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Enable bit, which enables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | | | ERR | OVF |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – MC1** Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

**Bit 4 – MC0** Match or Capture Channel x
This flag is set on a comparison match, or when the corresponding CCx register contains a valid capture value. This flag is set on the next CLK_TC_CNT cycle, and will generate an interrupt request if the corresponding Match or Capture Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In capture operation, this flag is automatically cleared when CCx register is read.

**Bit 1 – ERR** Error Interrupt Flag
This flag is set when a new capture occurs on a channel while the corresponding Match or Capture Channel x interrupt flag is set, in which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

**Bit 0 – OVF** Overflow Interrupt Flag
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if INTENCLR.OVF or INTENSET.OVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.8. Status

**Name:** STATUS
**Offset:** 0x0B
**Reset:** 0x01
**Property:** Read-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | CCBUFV1 | CCBUFV0 | PERBUFV | | SLAVE | STOP |
| Access | | | R/W | R/W | R/W | | R | R |
| Reset | | | 0 | 0 | 0 | | 0 | 1 |

**Bits 4, 5 – CCBUFV** Channel x Compare or Capture Buffer Valid
For a compare channel x, the bit x is set when a new value is written to the corresponding CCBUFx register.
The bit x is cleared by writing a '1' to it when CTRLB.LUPD is set, or it is cleared automatically by hardware on UPDATE condition.
For a capture channel x, the bit x is set when a valid capture value is stored in the CCBUFx register. The bit x is cleared automatically when the CCx register is read.

**Bit 3 – PERBUFV** Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. The bit is cleared by writing '1' to the corresponding location when CTRLB.LUPD is set, or automatically cleared by hardware on UPDATE condition. This bit is available only in 8-bit mode and will always read zero in 16- and 32-bit modes.

**Bit 1 – SLAVE** Client Status Flag
This bit is only available in 32-bit mode on the Client TC (i.e., TC1 and/or TC3). The bit is set when the associated Host TC (TC0 and TC2, respectively) is set to run in 32-bit mode.

**Bit 0 – STOP** Stop Status Flag
This bit is set when the TC is disabled, on a Stop command, or on an overflow/underflow condition when the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) is '1'.

| Value | Description |
|---|---|
| 0 | Counter is running. |
| 1 | Counter is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.9. Waveform Generation Control

**Name:** WAVE
**Offset:** 0x0C
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | WAVEGEN[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – WAVEGEN[1:0]** Waveform Generation Mode
These bits select the waveform generation operation. They affect the top value, as shown in Waveform Output Operations. They also control whether frequency or PWM waveform generation should be used. The waveform generation operations are explained in Waveform Output Operations.
These bits are not synchronized.

| Value | Name | Operation | Top Value | Output Waveform on Match | Output Waveform on Wraparound |
|---|---|---|---|---|---|
| 0x0 | NFRQ | Normal frequency | PER[1] / Max | Toggle | No action |
| 0x1 | MFRQ | Match frequency | CC0 | Toggle | No action |
| 0x2 | NPWM | Normal PWM | PER[1] / Max | Set | Clear |
| 0x3 | MPWM | Match PWM | CC0 | Set | Clear |

1) This depends on the TC mode: In 8-bit mode, the top value is the Period Value register (PER). In 16- and 32-bit mode it is the respective MAX value.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.3.10. Driver Control

**Name:** DRVCTRL
**Offset:** 0x0D
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | INVEN1 | INVEN0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 0, 1 – INVENx** Output Waveform x Invert Enable
Bit x of INVEN[1:0] selects inversion of the output or capture trigger input of channel x.

| Value | Description |
|---|---|
| 0 | Disable inversion of the WO[x] output and IO input pin. |
| 1 | Enable inversion of the WO[x] output and IO input pin. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.11. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0F
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Run in Debug Mode
This bit is not affected by a software Reset, and should not be changed by software while the TC is enabled.

| Value | Description |
|---|---|
| 0 | The TC is halted when the device is halted in debug mode. |
| 1 | The TC continues normal operation when the device is halted in debug mode. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 48.7.3.12. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC1 | CC0 | | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| Access | R | R | | R | R | R | R | R |
| Reset | 0 | 0 | | 0 | 0 | 0 | 0 | 0 |

**Bits 6, 7 – CCx** Compare/Capture Channel x Synchronization Busy
For details on CC channels number, refer to each TC feature list.
This bit is set when the synchronization of CCx between clock domains is started.
This bit is also set when the CCBUFx is written, and cleared on update condition. The bit is automatically cleared when the STATUS.CCBUFx bit is cleared.

**Bit 4 – COUNT** COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT between the clock domains is complete.
This bit is set when the synchronization of COUNT between clock domains is started.

**Bit 3 – STATUS** STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS between the clock domains is complete.
This bit is set when a '1' is written to the Capture Channel Buffer Valid status flags (STATUS.CCBUFVx) and the synchronization of STATUS between clock domains is started.

**Bit 2 – CTRLB** CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB between the clock domains is complete.
This bit is set when the synchronization of CTRLB between clock domains is started.

**Bit 1 – ENABLE** ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

**Bit 0 – SWRST** SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.13. Counter Value, 32-bit Mode

**Name:** COUNT
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

**Note:** Prior to any read access, this register must be synchronized by user by writing the according TC Command value to the Control B Set register (CTRLBSET.CMD=READSYNC).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – COUNT[31:0]** Counter Value
These bits contain the current counter value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.14. Channel x Compare/Capture Value, 32-bit Mode

**Name:** CCx
**Offset:** 0x1C + x*0x04 [x=0..1]
**Reset:** 0x00000000
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CC[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CC[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CC[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CC[31:0]** Channel x Compare/Capture Value
These bits contain the compare/capture value in 32-bit TC mode. In Match frequency (MFRQ) or Match PWM (MPWM) waveform operation (WAVE.WAVEGEN), the CC0 register is used as a period register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 48.7.3.15. Channel x Compare Buffer Value, 32-bit Mode

**Name:** CCBUFx
**Offset:** 0x30 + x*0x04 [x=0..1]
**Reset:** 0x00000000
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CCBUF[31:0]** Channel x Compare Buffer Value

These bits hold the value of the Channel x Compare Buffer Value. When the buffer valid flag is '1' and double buffering is enabled (CTRLBCLR.LUPD=1), the data from buffer registers will be copied into the corresponding CCx register under UPDATE condition (CTRLBSET.CMD=0x3), including the software update command.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 49. TCC – Timer/Counter for Control Applications

## 49.1. Overview

The device provides five instances of the Timer/Counter for Control applications (TCC) peripheral, TCC[4:0].

Each TCC instance consists of a counter, a prescaler, compare/capture channels and control logic. The counter can be set to count events or clock pulses. The counter together with the compare/capture channels can be configured to time stamp input events, allowing capture of frequency and pulse-width. It can also perform waveform generation, such as frequency generation and pulse-width modulation.

Waveform extensions are featured for motor control, ballast, LED, H-bridge, power converters, and other types of power control applications. They allow for low-side and high-side output with optional dead-time insertion. Waveform extensions can also generate a synchronized bit pattern across the waveform output pins. The fault options enable fault protection for safe and deterministic handling, disabling and/or shut down of external drivers.

**Note:** The TCC configurations, such as channel numbers and features, may be reduced for some of the TCC instances.

**Related Links**

TCC Configurations

## 49.2. Features

The following key features of the TCC:

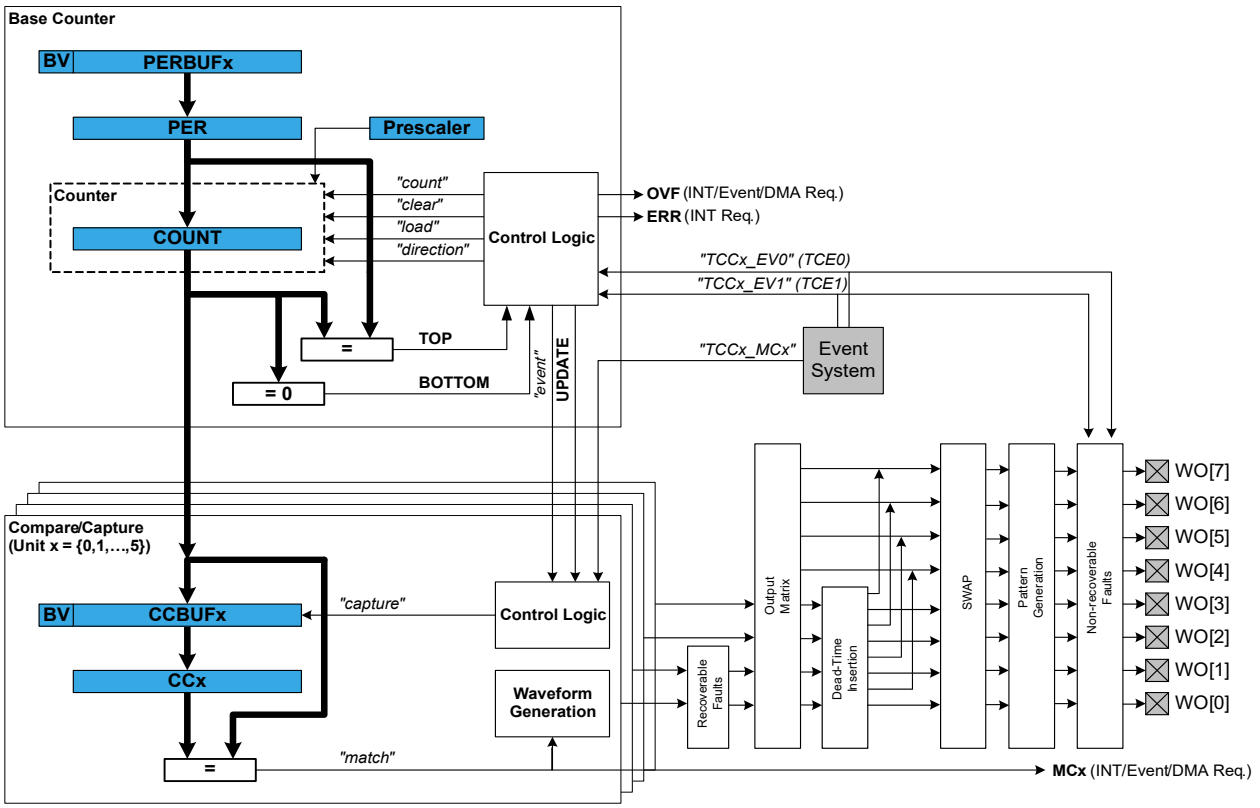
- Up to six Compare/Capture Channels (CC) with:
  - Double buffered period setting
  - Double buffered compare or capture channel
  - Circular buffer on period and compare channel registers
- Waveform Generation:
  - Frequency generation
  - Single-slope Pulse-Width Modulation (PWM)
  - Dual-slope PWM with half-cycle reload capability
- Input Capture:
  - Event capture
  - Frequency capture
  - Pulse-width capture
- Waveform Extensions:
  - Configurable distribution of compare channels outputs across port pins
  - Low-side and high-side output with programmable dead-time insertion
  - Waveform swap option with double buffer support
  - Pattern generation with double buffer support
  - Dithering support
- Fault Protection for Safe Disabling of Drivers:
  - Two recoverable fault sources
  - Two non-recoverable fault sources


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - Debugger can be a source of non-recoverable fault
- Input Events:
  - Two input events (EVx) for counter
  - One input event (MCx) for each channel
- Output Events:
  - Three output events (Count, re-trigger, and overflow) are available for counter
  - One compare match/input capture event output for each channel
- Interrupts:
  - Overflow and re-trigger interrupt
  - Compare match/input capture interrupt
  - Interrupt on fault detection

## 49.3. Block Diagram

**Figure 49-1.** Timer/Counter for Control Applications - Block Diagram




## 49.4. Signal Description

| Pin Name | Type | Description |
|---|---|---|
| TCC/WO[0] | Digital output | Compare channel 0 waveform output |
| TCC/WO[1] | Digital output | Compare channel 1 waveform output |
| ... | ... | ... |
| TCC/WO[7] | Digital output | Compare channel 7 waveform output |

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
I/O Multiplexing and Considerations

## 49.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 49.5.1. I/O Lines

In order to use the I/O lines of this peripheral, the I/O pins must be configured using the I/O Pin Controller (PORT).

**Related Links**
PORT - I/O Pin Controller

### 49.5.2. Power Management

This peripheral can continue to operate in any Sleep mode where its source clock is running. The interrupts can wake up the device from Sleep modes. Events connected to the event system can trigger other operations in the system without exiting Sleep modes.

### 49.5.3. Clocks

The TCC bus clocks (CLK_TCCx_APB) can be enabled and disabled in the Main Clock module. The default state of CLK_TCCx_APB can be found in the Peripheral Clock Masking section (see the Related Links below).

A generic clock (GCLK_TCCx) is required to clock the TCC. This clock must be configured and enabled in the generic clock controller before using the TCC.

The generic clocks (GCLK_TCCx) are asynchronous to the bus clock (CLK_TCCx_APB). Due to this asynchronicity, writing certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

**Related Links**
Peripheral Clock Masking
GCLK - Generic Clock Controller

### 49.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). In order to use DMA requests with this peripheral the DMAC must be configured first. Refer to *DMAC – Direct Memory Access Controller* for details.

**Related Links**
DMAC – Direct Memory Access Controller

### 49.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller

### 49.5.6. Events

The events of this peripheral are connected to the Event System.

**Related Links**
Event System (EVSYS)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will halt normal operation. This peripheral can be forced to continue operation during debugging - refer to the Debug Control (DBGCTRL) register for details.

Refer to DBGCTRL register for details.

### 49.5.8. Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag register (INTFLAG)
- Status register (STATUS)
- Period and Period Buffer registers (PER, PERBUF)
- Compare/Capture and Compare/Capture Buffer registers (CCx, CCBUFx)
- Control Waveform register (WAVE)
- Pattern Generation Value and Pattern Generation Value Buffer registers (PATT, PATTBUF)

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

### 49.5.9. Analog Connections

Not applicable.

## 49.6. Functional Description

### 49.6.1. Principle of Operation

The following definitions are used throughout the documentation:

**Table 49-1.** Timer/Counter for Control Applications - Definitions

| Name | Description |
|---|---|
| TOP | The counter reaches TOP when it becomes equal to the highest value in the count sequence. The TOP value can be the same as Period (PER) or the Compare Channel 0 (CC0) register value depending on the Waveform Generator mode in Waveform Output Generation Operations. |
| ZERO | The counter reaches ZERO when it contains all zeroes. |
| MAX | The counter reaches maximum when it contains all ones. |
| UPDATE | The timer/counter signals an update when it reaches ZERO or TOP, depending on the direction settings. |
| Timer | The timer/counter clock control is handled by an internal source. |
| Counter | The clock control is handled externally (e.g., counting external events). |
| CC | For compare operations, the CC are referred to as "compare channels."<br>For capture operations, the CC are referred to as "capture channels." |

Each TCC instance has up to six compare/capture channels (CCx).

The Counter register (COUNT), Period registers with Buffer (PER and PERBUF), and Compare and Capture registers with buffers (CCx and CCBUFx) are 16-bit or 24-bit registers, depending on each TCC instance. Each Buffer register has a Buffer Valid (BUFV) flag that indicates when the buffer contains a new value.

Under normal operation, the counter value is continuously compared to the TOP or ZERO value to determine whether the counter has reached TOP or ZERO. In either case, the TCC can generate


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

interrupt requests or generate events for the Event System. In Waveform Generator mode, these comparisons are used to set the waveform period or pulse width.

A prescaled generic clock (GCLK_TCCx) and events from the event system can be used to control the counter. The event system is also used as a source to the input capture.

The Recoverable Fault Unit enables event controlled waveforms by acting directly on the generated waveforms of the TCC compare channels output. These events can restart, halt the timer/counter period, shorten the output pulse active time, or disable waveform output as long as the fault condition is present. This can typically be used for current sensing regulation, and zero-crossing and demagnetization re-triggering.

The MCE0 and MCE1 asynchronous event sources are shared with the recoverable fault unit. Only asynchronous events are used internally when fault unit extension is enabled. For further details on how to configure asynchronous events routing, refer to *EVSYS – Event System*.

Recoverable fault sources can be filtered and/or windowed to avoid false triggering, for example from I/O pin glitches, by using digital filtering, input blanking, and qualification options. See also Recoverable Faults.

In order to support applications with different types of motor control, ballast, LED, H-bridge, power converter, and other types of power switching applications, the following independent units are implemented in some of the TCC instances as optional and successive units:

- Recoverable faults and non-recoverable faults
- Output matrix
- Dead-time insertion
- Swap
- Pattern generation

See also Figure 49-1.

The output matrix (OTMX) can distribute and route out the TCC waveform outputs across the port pins in different configurations, each optimized for different application types. The Dead-Time Insertion (DTI) unit splits the four lower OTMX outputs into two non-overlapping signals: the non-inverted Low Side (LS) and inverted High Side (HS) of the waveform output with optional dead-time insertion between LS and HS switching. The SWAP unit can swap the LS and HS pin outputs, and can be used for fast decay motor control.

The pattern generation unit can be used to generate synchronized waveforms with constant logic level on TCC UPDATE conditions. This is useful for easy stepper motor and full bridge control.

The non-recoverable fault module enables event controlled fault protection by acting directly on the generated waveforms of the timer/counter compare channel outputs. When a non-recoverable fault condition is detected, the output waveforms are forced to a preconfigured value that is safe for the application. This is typically used for instant and predictable shut down and disabling high current or voltage drives.

The count event sources (TCE0 and TCE1) are shared with the non-recoverable fault extension. The events can be optionally filtered. If the filter options are not used, the non-recoverable faults provide an immediate asynchronous action on waveform output, even for cases where the clock is not present. For further details on how to configure asynchronous events routing, refer to section *EVSYS – Event System*.

**Related Links**

Event System (EVSYS)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 49.6.2. Basic Operation

### 49.6.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the TCC is disabled (CTRLA.ENABLE = 0):

- The Control A (CTRLA) register, except Run Standby (RUNSTDBY), Enable (ENABLE) and Software Reset (SWRST) bits
- The Recoverable Fault n Control registers (FCTRLA and FCTRLB)
- The Waveform Extension Control register (WEXCTRL)
- The Drive Control register (DRVCTRL)
- The Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written simultaneously as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'. Enable-protection is denoted by the 'Enable-Protected' property in the register description.

Before enabling the TCC, follow these steps to configure it:

1. Enable the TCC bus clock (CLK_TCCx_APB).
2. If Capture mode is required, enable the channel in Capture mode by writing a '1' to the Capture Enable bit in the Control A register (CTRLA.CPTEN).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER).
2. Select Prescaler Synchronization setting in the Control A register (CTRLA.PRESCSYNC).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR) to '1'.
4. Select the Waveform Generation operation in the WAVE register (WAVE.WAVEGEN).
5. Select the Waveform Output Polarity in the WAVE register (WAVE.POL).
6. The waveform output can be inverted for the individual channels using the Waveform Output Invert Enable bit group in the Driver register (DRVCTRL.INVEN).

### 49.6.2.2. Enabling, Disabling, and Resetting

The TCC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The TCC is disabled by writing a zero to CTRLA.ENABLE.

The TCC is reset by writing '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the TCC, except DBGCTRL, will be reset to their initial state, and the TCC will be disabled. Refer to Control A (CTRLA) register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

### 49.6.2.3. Prescaler Selection

The GCLK_TCCx clock is fed into the internal prescaler.

The prescaler consists of a counter that counts up to the selected prescaler value, whereupon the output of the prescaler toggles.

If the prescaler value is higher than one, the Counter Update condition can be optionally executed on the next GCLK_TCCx clock pulse or the next prescaled clock pulse. For further details, refer to the Prescaler (CTRLA.PRESCALER) and Counter Synchronization (CTRLA.PRESYNC) descriptions.

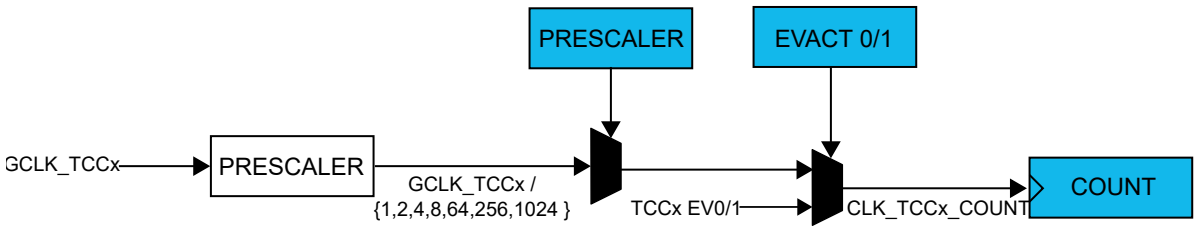
Prescaler outputs from 1 to 1/1024 are available. For a complete list of available prescaler outputs, see the register description for the Prescaler bit group in the Control A register (CTRLA.PRESCALER).

**Note:** When counting events, the prescaler is bypassed.

The joint stream of prescaler ticks and event action ticks is called CLK_TCCx_COUNT.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-2.** Prescaler




### 49.6.2.4. Counter Operation

Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCCx_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

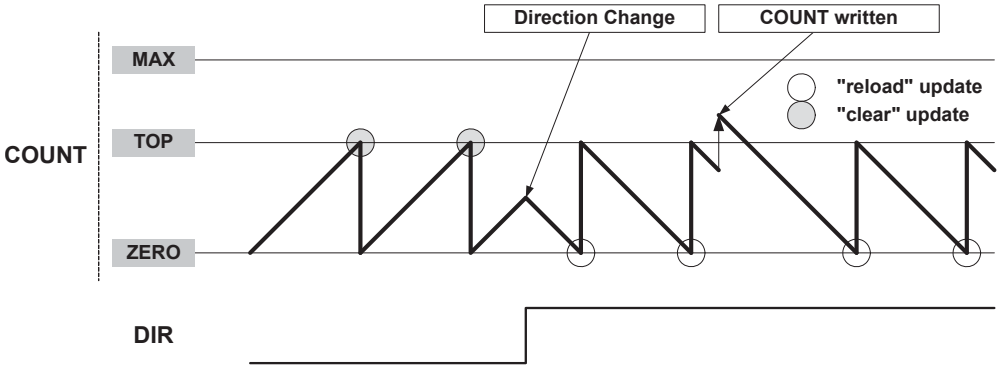
The counting direction is set by the Direction bit in the Control B register (CTRLB.DIR). If the bit is zero, it's counting up and one if counting down.

The counter will count up or down for each tick (clock or event) until it reaches TOP or ZERO. When it's counting up and TOP is reached, the counter will be set to zero at the next tick (overflow) and the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) will be set. When down-counting, the counter is reloaded with the TOP value when ZERO is reached (underflow), and INTFLAG.OVF is set.

**Note:** When the TCC is counting down, the COUNT register must be initialized to the TOP value (PER or CC0 value depending on the mode).

INTFLAG.OVF can be used to trigger an interrupt, or an event. An overflow/underflow occurrence (i.e., a compare match with TOP/ZERO) will stop counting if the One-Shot bit in the Control B register is set (CTRLBSET.ONESHOT). The One-Shot feature is explained in the Additional Features section.

**Figure 49-3.** Counter Operation




Users can change the counter value (by writing directly in the COUNT register) even when the counter is running. The COUNT value will always be ZERO or TOP, depending on direction set by CTRLBSET.DIR or CTRLBCLR.DIR, when starting the TCC, unless a different value has been written to it, or the TCC has been stopped at a value other than ZERO. The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation.

#### Stop Command

A stop command can be issued from software by using TCC Command bits in the Control B Set register (CTRLBSET.CMD = 0x2, STOP).

When a stop is detected while the counter is running, the counter will maintain its current value. If the waveform generation (WG) is used, all waveforms are set to a state defined in Non-


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Recoverable State x Output Enable bit and Non- Recoverable State x Output Value bit in the Driver Control register (DRVCTRL.NREx and DRVCTRL.NRVx), and the Stop bit in the Status register is set (STATUS.STOP).

**Pause Event Action**

A pause command can be issued when the stop event action is configured in the Input Event Action 1 bits in Event Control register (EVCTRL.EVACT1 = 0x3, STOP).

When a pause is detected, the counter can stop immediatly maintaining its current value and all waveforms keep their current state, as long as a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0 = 0x3, START).

**Re-Trigger Command and Event Action**

A re-trigger command can be issued from software by using TCC Command bits in Control B Set register (CTRLBSET.CMD = 0x1, RETRIGGER), or from event when the re-trigger event action is configured in the Input Event 0/1 Action bits in Event Control register (EVCTRL.EVACTn = 0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR or CTRLBCLR.DIR). The Re-Trigger bit in the Interrupt Flag Status and Clear register will be set (INTFLAG.TRG). It is also possible to generate an event by writing a '1' to the Re-Trigger Event Output Enable bit in the Event Control register (EVCTRL.TRGEO). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in COUNT.

**Note:**
When a re-trigger event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACTn = 0x1, RETRIGGER), enabling the counter will not start the counter. The counter will start on the next incoming event and restart on corresponding following event.

**Start Event Action**

The start action can be selected in the Event Control register (EVCTRL.EVACT0 = 0x3, START) and can start the counting operation when previously stopped. The event has no effect if the counter is already counting. When the module is enabled, the counter operation starts when the event is received or when a re-trigger software command is applied.

**Note:**
When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0 = 0x3, START), enabling the counter will not start the counter. The counter will start on the next incoming event, but it will not restart on subsequent events.

**Count Event Action**

The TCC can count events. When an event is received, the counter increases or decreases the value, depending on direction settings (CTRLBSET.DIR or CTRLBCLR.DIR).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0 = 0x5, COUNT).

**Direction Event Action**

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1 = 0x2, DIR). When this event is used, the asynchronous event path specified in the event system must be configured or selected. The direction event action can be used to control the direction of the counter operation, depending on external events level. When received, the event level overrides the Direction settings (CTRLBSET.DIR or CTRLBCLR.DIR) and the direction bit value is updated accordingly.

**Increment Event Action**


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0 = 0x4, INC) and can change the Counter state when an event is received. When the TCE0 event (TCCx_EV0) is received, the counter increments, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

**Decrement Event Action**

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1 = 0x4, DEC) and can change the Counter state when an event is received. When the TCE1 (TCCx_EV1) event is received, the counter decrements, whatever the direction setting (CTRLBSET.DIR or CTRLBCLR.DIR) is.

**Non-recoverable Fault Event Action**

Non-recoverable fault actions can be selected in the Event Control register (EVCTRL.EVACTn = 0x7, FAULT). When received, the counter will be stopped and the output of the compare channels is overridden according to the Driver Control register settings (DRVCTRL.NREx and DRVCTRL.NRVx). TCE0 and TCE1 must be configured as asynchronous events.

**Event Action Off**

If the event action is disabled (EVCTRL.EVACTn = 0x0, OFF), enabling the counter will also start the counter.

**Related Links**

One-Shot Operation

#### 49.6.2.5. Compare Operations

By default, the Compare/Capture channel is configured for compare operations. To perform capture operations, it must be re-configured.

When using the TCC with the Compare/Capture Value registers (CCx) for compare operations, the counter value is continuously compared to the values in the CCx registers. This can be used for timer or for waveform operation.

The Channel x Compare/Capture Buffer Value (CCBUFx) registers provide double buffer capability. The double buffering synchronizes the update of the CCx register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD=0x3, UPDATE). For further details, refer to Double Buffering. The synchronization prevents the occurrence of odd-length, non-symmetrical pulses and ensures glitch-free output.

##### 49.6.2.5.1. Waveform Output Generation Operations

The compare channels can be used for waveform generation on output port pins. To make the waveform available on the connected pin, the following requirements must be fulfilled:

1. Choose a Waveform Generation mode in the Waveform Generation Operation bit in Waveform register (WAVE.WAVEGEN).
2. Optionally invert the waveform output WO[x] by writing the corresponding Waveform Output x Inversion bit in the Driver Control register (DRVCTRL.INVENx).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* for details.
   **Note:** Event must not be used when the compare channel is set in waveform output operating mode.

The counter value is continuously compared with each CCx value. On a comparison match, the Match or Capture Channel x bit in the Interrupt Flag Status and Clear register (INTFLAG.MCx) will be set on the next zero-to-one transition of CLK_TCC_COUNT (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCx and/or EVCTRL.MCEOx is '1'. Both interrupt and event can be generated simultaneously.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

There are seven waveform configurations for the Waveform Generation Operation bit group in the Waveform register (WAVE.WAVEGEN). This will influence how the waveform is generated and impose restrictions on the top value. The configurations are:

- Normal Frequency (NFRQ)
- Match Frequency (MFRQ)
- Normal Pulse-Width Modulation (NPWM)
- Dual-slope, interrupt/event at TOP (DSTOP)
- Dual-slope, interrupt/event at ZERO (DSBOTTOM)
- Dual-slope, interrupt/event at Top and ZERO (DSBOTH)
- Dual-slope, critical interrupt/event at ZERO (DSCRITICAL)

When using MFRQ configuration, the TOP value is defined by the CC0 register value. For the other waveform operations, the TOP value is defined by the Period (PER) register value.

For dual-slope waveform operations, the update time occurs when the counter reaches ZERO. For the other Waveforms Generation modes, the update time occurs on counter wraparound, on overflow, underflow, or re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different operation modes.

**Table 49-2.** Counter Update and Overflow Event/interrupt Conditions

| Name | Operation | TOP | Update | Output Waveform | | OVFIF/Event | |
|---|---|---|---|---|---|---|---|
| | | | | On Match | On Update | Up | Down |
| NFRQ | Normal Frequency | PER | TOP/ ZERO | Toggle | Stable | TOP | ZERO |
| MFRQ | Match Frequency | CC0 | TOP/ ZERO | Toggle | Stable | TOP | ZERO |
| NPWM | Single-slope PWM | PER | TOP/ ZERO | See section 'Output Polarity' below | | TOP | ZERO |
| DSCRITICAL | Dual-slope PWM | PER | ZERO | | | - | ZERO |
| DSBOTTOM | Dual-slope PWM | PER | ZERO | | | - | ZERO |
| DSBOTH | Dual-slope PWM | PER | TOP(1) & ZERO | | | TOP | ZERO |
| DSTOP | Dual-slope PWM | PER | ZERO | | | TOP | – |

1. The UPDATE condition on TOP only will occur when circular buffer is enabled for the channel.

**Related Links**

Circular Buffer

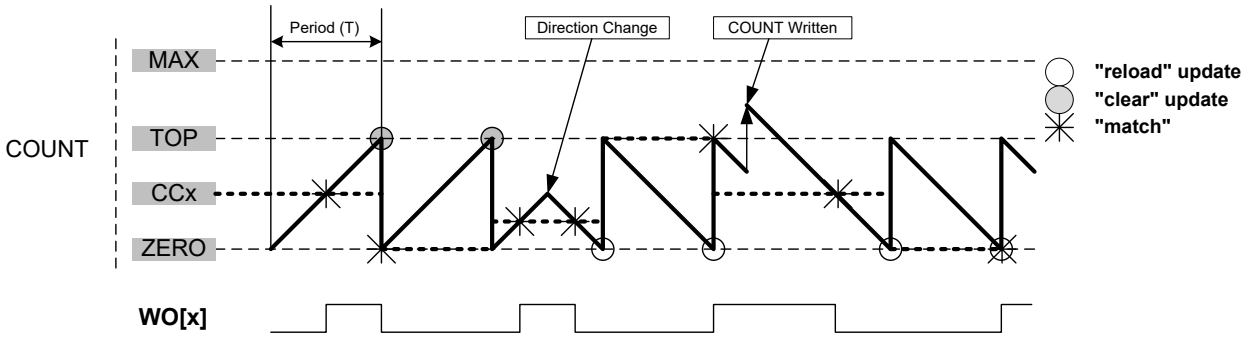
PORT - I/O Pin Controller

#### 49.6.2.5.2. Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[x]) is toggled on each compare match between COUNT and CCx, and the corresponding Match or Capture Channel x Interrupt Flag (EVCTRL.MCEOx) will be set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

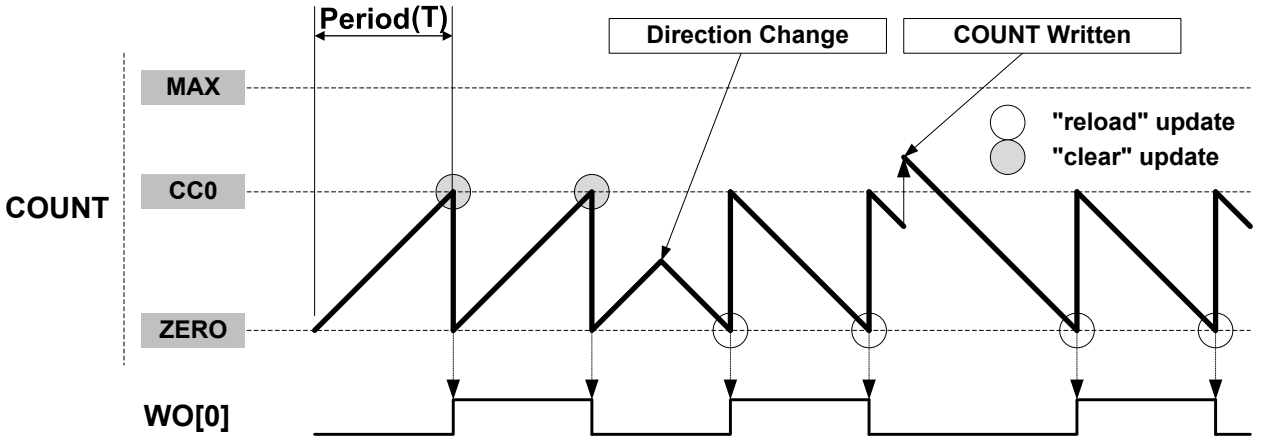
**Figure 49-4.** Normal Frequency Operation




#### 49.6.2.5.3. Match Frequency (MFRQ)

For Match Frequency generation, the period time (T) is controlled by CC0 register instead of PER. WO[0] toggles on each update condition.

**Figure 49-5.** Match Frequency Operation




#### 49.6.2.5.4. Normal Pulse-Width Modulation (NPWM)

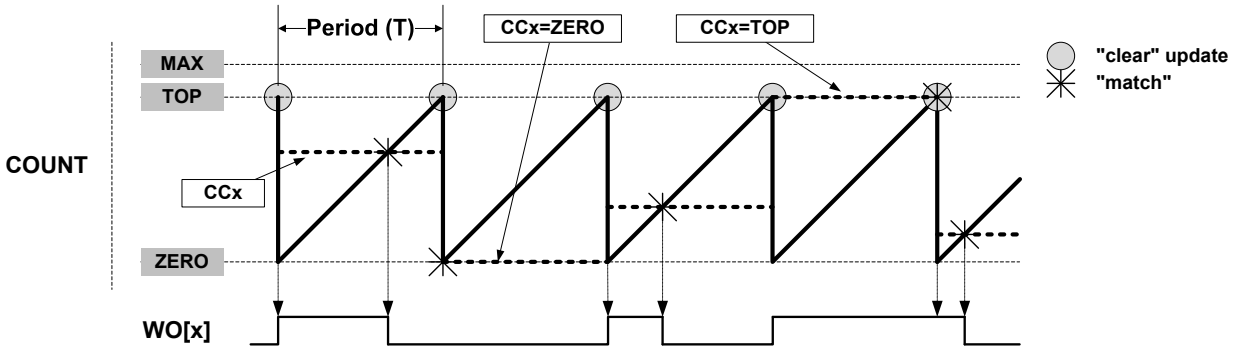
NPWM uses single-slope PWM generation.

#### 49.6.2.5.5. Single-Slope PWM Operation

For single-slope PWM generation, the period time (T) is controlled by Top value, and CCx controls the duty cycle of the generated waveform output. When up-counting, the WO[x] is set at start or compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCx register values. When down-counting, the WO[x] is cleared at start or compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCx register values.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-6.** Single-Slope PWM Operation




The following equation calculates the exact resolution for a single-slope PWM ($R_{PWM_SS}$) waveform:

$$R_{PWM_SS} = \frac{\log(TOP+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency ($f_{GCLK_TCCx}$), and can be calculated by the following equation:

$$f_{PWM_SS} = \frac{f_{GCLK_TCCx}}{N(TOP+1)}$$

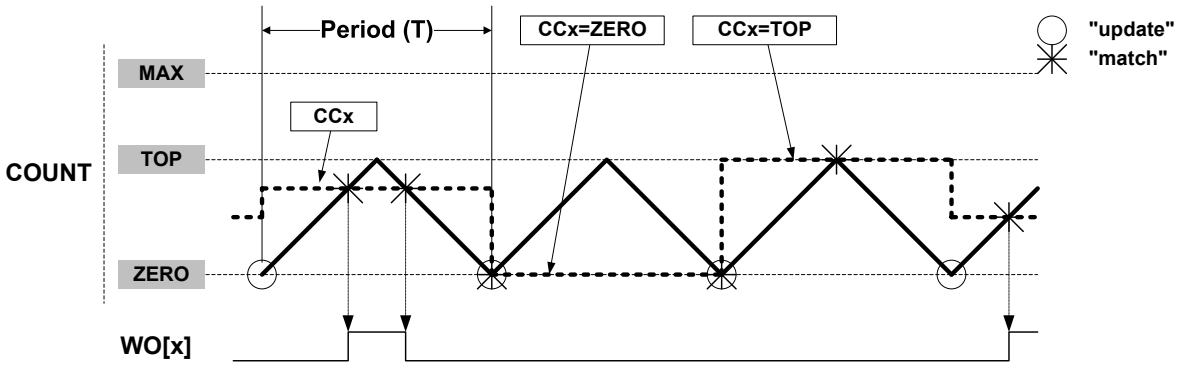
Where N represents the prescaler divider used (1, 2, 4, 8, 16, 64, 256, 1024).

### 49.6.2.5.6. Dual-Slope PWM Generation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCx control the duty cycle of the generated waveform output. The figure below shows how the counter repeatedly counts from ZERO to PER and then from PER to ZERO. The waveform generator output is set on compare match when up-counting, and cleared on compare match when down-counting. An interrupt and/or event is generated on TOP (when counting upwards) and/or ZERO (when counting up or down).

In DSBOTH operation, the circular buffer must be enabled to enable the update condition on TOP.

**Figure 49-7.** Dual-Slope Pulse Width Modulation




Using dual-slope PWM results in a lower maximum operation frequency compared to single-slope PWM generation. The period (TOP) defines the PWM resolution. The minimum resolution is 1 bit (TOP=0x00000001).

The following equation calculates the exact resolution for dual-slope PWM ($R_{PWM_DS}$):

$$R_{PWM_DS} = \frac{\log(PER+1)}{\log(2)}.$$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The PWM frequency $f_{PWM_DS}$ depends on the period setting (TOP) and the peripheral clock frequency $f_{GCLK_TCCx}$, and can be calculated by the following equation:

$$f_{PWM_DS} = \frac{f_{GCLK_TCCx}}{2N \cdot PER}$$

*N* represents the prescaler divider used. The waveform generated will have a maximum frequency of half of the TCC clock frequency ($f_{GCLK_TCCx}$) when TOP is set to 0x00000001 and no prescaling is used.

The pulse width ($P_{PWM_DS}$) depends on the compare channel (CCx) register value and the peripheral clock frequency ($f_{GCLK_TCCx}$), and can be calculated by the following equation:

$$P_{PWM_DS} = \frac{2N \cdot (TOP - CCx)}{f_{GCLK_TCCx}}$$

*N* represents the prescaler divider used.

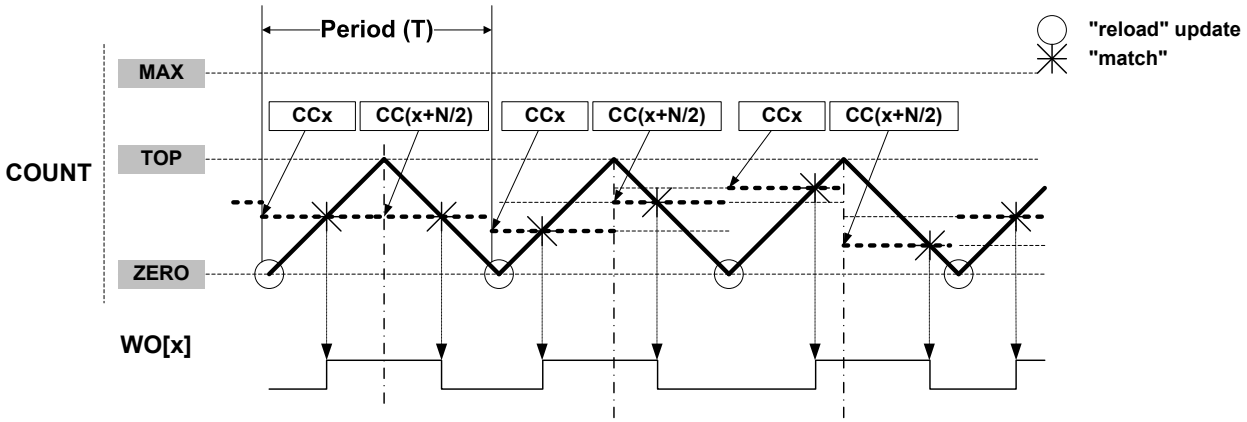
**Note:** In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the CCx MSB bit defines the ramp on which the CCx Match interrupt or event is generated. (Rising if CCx[MSB] = 0, falling if CCx[MSB] = 1.)

**Related Links**
Circular Buffer

### 49.6.2.5.7. Dual-Slope Critical PWM Generation

Critical mode generation allows generation of non-aligned centered pulses. In this mode, the period time is controlled by PER while CCx control the generated waveform output edge during up-counting and CC(x+CC_NUM/2) control the generated waveform output edge during down-counting.

**Figure 49-8.** Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)




### 49.6.2.5.8. Output Polarity

The polarity (WAVE.POLx) is available in all waveform output generation. In single-slope and dual-slope PWM operation, it is possible to invert the pulse edge alignment individually on start or end of a PWM cycle for each compare channels. The table below shows the waveform output set/clear conditions, depending on the settings of timer/counter, direction, and polarity.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 49-3.** Waveform Generation Set/Clear Conditions

| Waveform Generation Operation | DIR | POLx | Waveform Generation Output Update | |
|---|---|---|---|---|
| | | | Set | Clear |
| Single-Slope PWM | 0 | 0 | Timer/counter matches TOP | Timer/counter matches CCx |
| | | 1 | Timer/counter matches CC | Timer/counter matches TOP |
| | 1 | 0 | Timer/counter matches CC | Timer/counter matches ZERO |
| | | 1 | Timer/counter matches ZERO | Timer/counter matches CC |
| Dual-Slope PWM | x | 0 | Timer/counter matches CC when counting up | Timer/counter matches CC when counting down |
| | | 1 | Timer/counter matches CC when counting down | Timer/counter matches CC when counting up |

In Normal and Match Frequency, the WAVE.POLx value represents the initial state of the waveform output.

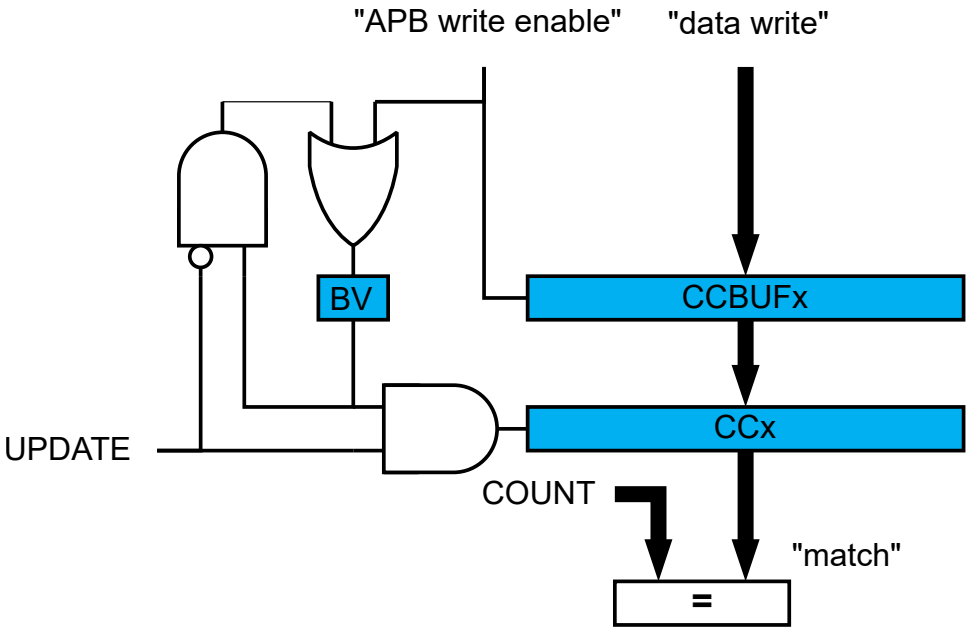
#### 49.6.2.6. Double Buffering

The Pattern (PATT), Period (PER) and Compare Channels (CCx) registers are all double buffered. Each buffer register has a buffer valid (PATTBUFV, PERBUFV and CCBUFVx) bit in the STATUS register, which indicates that the Buffer register contains a valid value that can be copied into the corresponding register. As long as the respective Buffer Valid Status flag (PATTBUFV, PERBUFV or CCBUFVx) are set to '1', the related SYNCBUSY bits are set (SYNCBUSY.PATT, SYNCBUSY.PER or SYNCBUSY.CCx), a write to the respective PATT/PATTBUF, PER/PERBUF or CCx/CCBUFx registers will generate a PAC error, and read access to the respective PATT, PER or CCx register is invalid.

When the Buffer Valid Flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLB register is set to '0', (writing CTRLBCLR.LUPD to '1'), double buffering is enabled: the data from buffer registers will be copied into the corresponding register under hardware UPDATE conditions, then the Buffer Valid flags bit in the STATUS register are automatically cleared by hardware.
**Note:** Software update command (CTRLBSET.CMD=0x3) act independently of LUPD value.

A compare register is double buffered as in the following figure.

**Figure 49-9.** Compare Channel Double Buffering





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Both the registers (PATT/PER/CCx) and corresponding Buffer registers (PATTBUFPERBUF/CCBUFx) are available in the I/O register map, and the double buffering feature is not mandatory. The double buffering is disabled by writing a '1' to CTRLSET.LUPD.

**Note:** In NFRQ, MFRQ or PWM Down-Counting Counter mode (CTRLBSET.DIR=1), when double buffering is enabled (CTRLBCLR.LUPD=1), PERBUF register is continuously copied into the PER independently of update conditions.

### Changing the Period

The counter period can be changed by writing a new Top value to the Period register (PER or CC0, depending on the Waveform Generation mode), any period update on registers (PER or CCx) is effective after the synchronization delay, whatever double buffering enabling is.

**Figure 49-10.** Unbuffered Single-Slope Up-Counting Operation

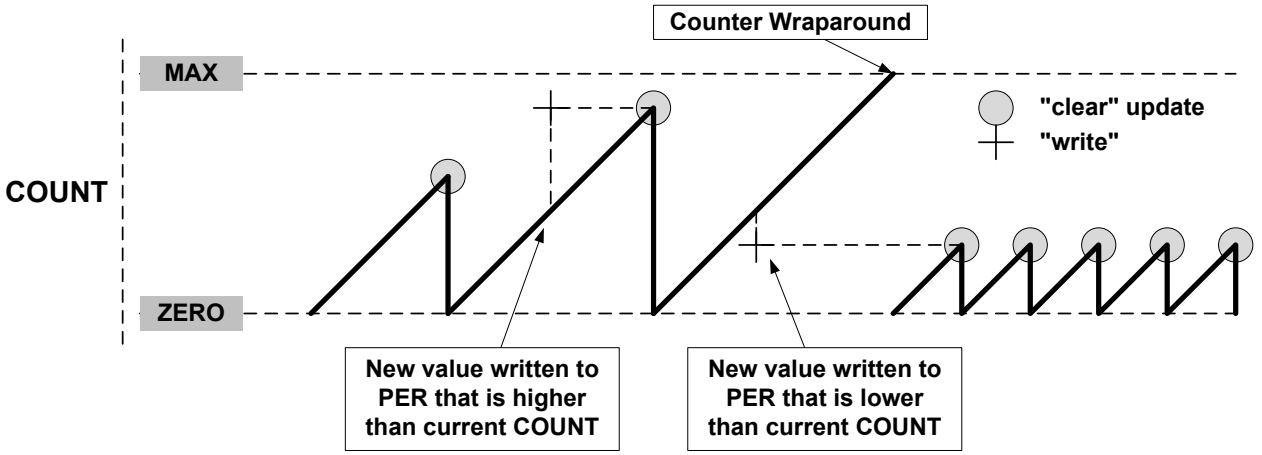
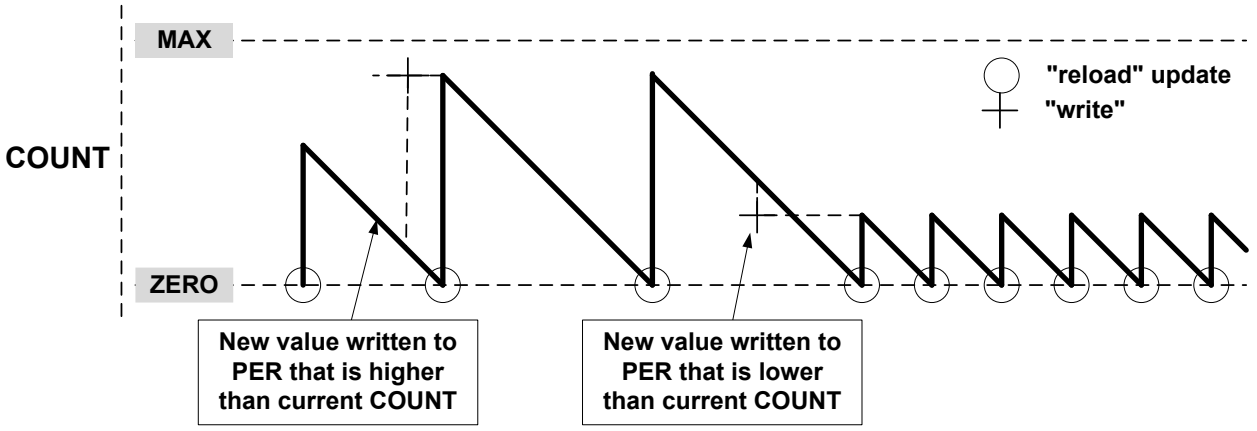


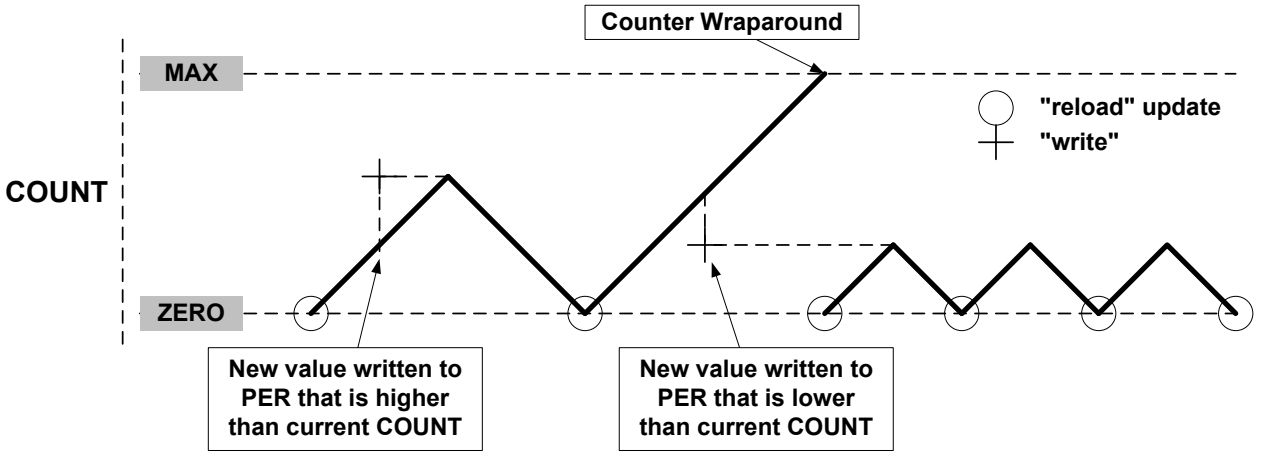

**Figure 49-11.** Unbuffered Single-Slope Down-Counting Operation




A counter wraparound can occur in any operation mode when up-counting without buffering, see Figure 49-10. COUNT and TOP are continuously compared, so when a new value that is lower than the current COUNT is written to TOP, COUNT will wrap before a compare match.

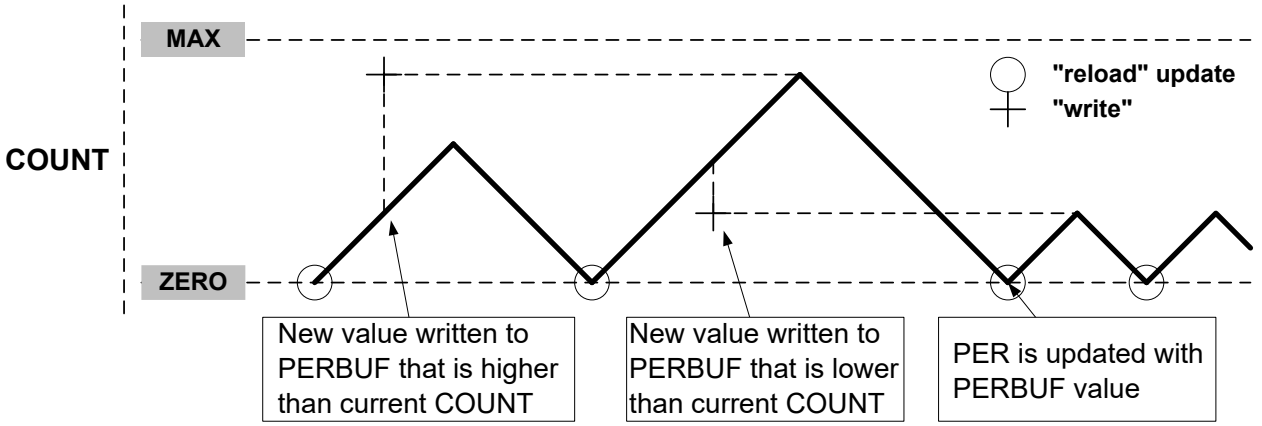

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-12.** Unbuffered Dual-Slope Operation




When double buffering is used, the buffer can be written at any time and the counter will still maintain correct operation. The period register is always updated on the update condition, as shown in Figure 49-13. This prevents wraparound and the generation of odd waveforms.

**Figure 49-13.** Changing the Period Using Buffering




### 49.6.2.7. Capture Operations

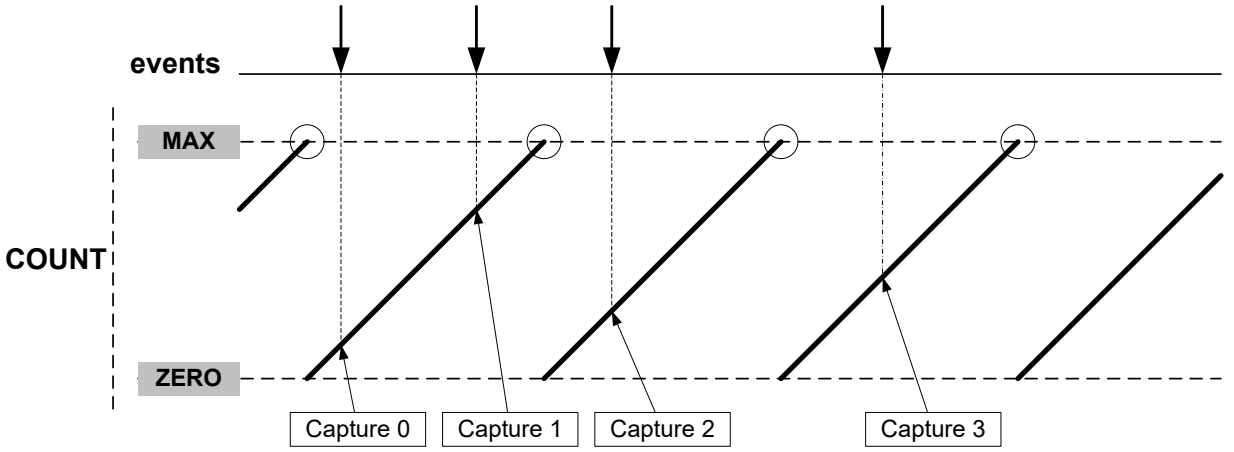
To enable and use capture operations, the Match or Capture Channel x Event Input Enable bit in the Event Control register (EVCTRL.MCEIx) must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel x Enable bit in the Control A register (CTRLA.CPTENx) before capturing can be performed.

#### Event Capture Action

The compare/capture channels can be used as input capture channels to capture events from the Event System, and give them a timestamp. The following figure shows four capture events for one capture channel. Event system channels must be configured to operate in asynchronous mode when used for capture operations.

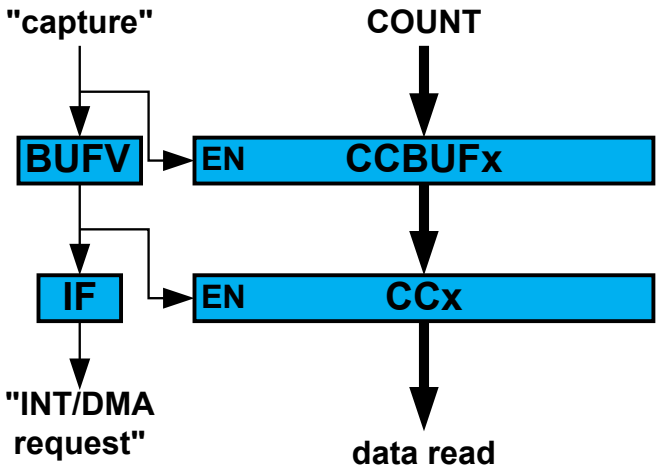

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-14.** Input Capture Timing




For input capture, the Buffer register and the corresponding CCx act like a FIFO. When CCx is empty or read, any content in CCBUFx is transferred to CCx. The Buffer Valid flag is passed to set the CCx Interrupt flag (IF) and generate the optional interrupt, event, or DMA request. The CCBUFx register value cannot be read, all captured data must be read from the CCx register.

**Figure 49-15.** Capture Double Buffering




The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the Capture Buffer Valid flag (STATUS.CCBUFV) is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

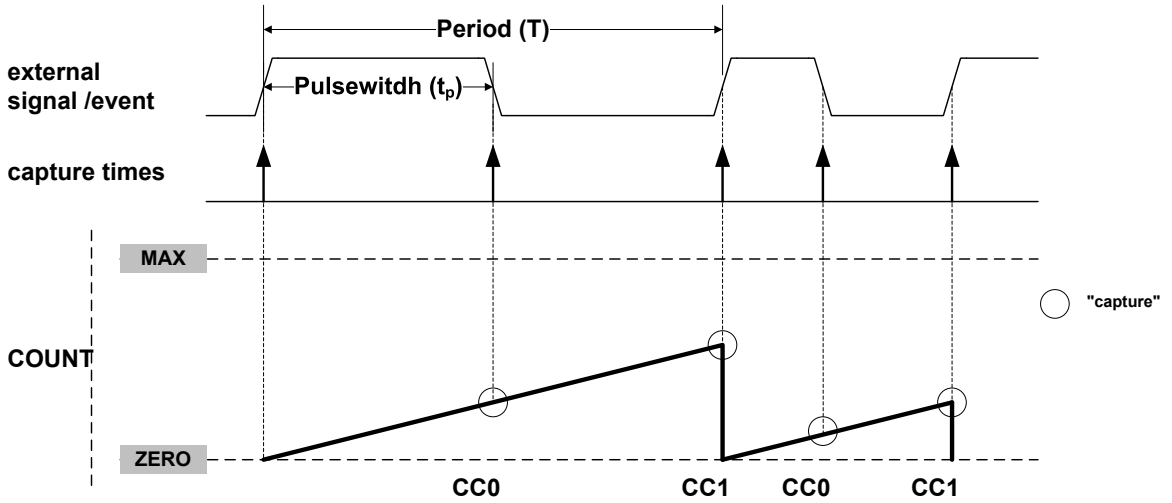
**Period and Pulse-Width (PPW) Capture Action**

The TCC can perform two input captures and restart the counter on one of the edges. This enables the TCC to measure the pulse-width and period and to characterize the frequency *f* and *dutyCycle* of an input signal, as shown below:

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-16.** PWP Capture




Selecting PWP or PPW in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. When using PPW event action, period *T* will be captured into CC0 and the pulse-width $t_p$ into CC1. The PWP (Pulse-width and Period) event action offers the same functionality, but *T* will be captured into CC1 and $t_p$ into CC0.

The Timer/Counter Event x Invert Enable bit in Event Control register (EVCTRL.TCEINVx) is used for event source x to select whether the wraparound should occur on the rising edge or the falling edge. If EVCTRL.TCEINVx=1, the wraparound will happen on the falling edge.

The corresponding capture is done only if the channel is enabled in Capture mode (CTRLA.CPTENx=1). If not, the capture action will be ignored and the channel will be enabled in compare mode of operation. When only one of these channel is required, the other channel can be used for other purposes.

The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the INTFLAG.MCx is still set, the new timestamp will not be stored and INTFLAG.ERR will be set.

**Note:** When up-counting (CTRLBSET.DIR=0), counter values lower than 1 cannot be captured in Capture Minimum mode (FCTRLn.CAPTURE=CAPTMIN). To capture the full range including value 0, the TCC must be configured in Down-counting mode (CTRLBSET.DIR=0).

**Note:** In dual-slope PWM operation, and when TOP is lower than MAX/2, the CCx MSB captures the CTRLB.DIR state to identify the ramp on which the capture has been done. For rising ramps CCx[MSB] is zero, for falling ramps CCx[MSB]=1.

## 49.6.3. Additional Features

### 49.6.3.1. One-Shot Operation

When one-shot is enabled, the counter automatically stops on the next Counter Overflow or Underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP) is set and the waveform outputs are set to the value defined by DRVCTRL.NREx and DRVCTRL.NRVx.

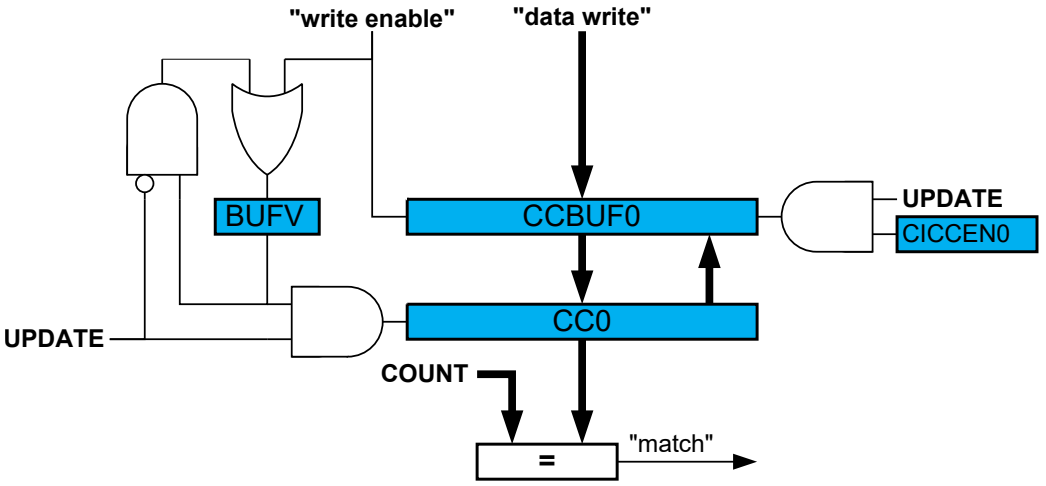
One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT) and disabled by writing a '1' to CTRLBCLR.ONESHOT. When enabled, the TCC will count until an overflow or underflow occurs and stop counting. The one-shot operation can be restarted by a re-trigger software command, a re-trigger event or a start event. When the counter restarts its operation, STATUS.STOP is automatically cleared.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.6.3.2. Circular Buffer

The Period register (PER) and the Compare Channels register (CC0 toCC5) support circular buffer operation. When circular buffer operation is enabled, the PER or CCx values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

**Figure 49-17.** Circular Buffer on Channel 0




### 49.6.3.3. Dithering Operation

The TCC supports dithering on Pulse-width or Period on a 16, 32, or 64 PWM cycles frame.

Dithering consists in adding some extra clocks cycles in a frame of several PWM cycles, and can improve the accuracy of the *average* output pulse width and period. The extra clock cycles are added on some of the compare match signals, one at a time, through a "blue noise" process that minimizes the flickering on the resulting dither patterns.

Dithering is enabled by writing the corresponding configuration in the Enhanced Resolution bits in CTRLA register (CTRLA.RESOLUTION):

- DITH4 enable dithering every 16 PWM frames
- DITH5 enable dithering every 32 PWM frames
- DITH6 enable dithering every 64 PWM frames

The DITHERCY bits of COUNT, PER, and CCx registers define the number of extra cycles to add into the frame (DITHERCY bits from the respective COUNT, PER or CCx registers). The remaining bits of the COUNT, PER, CCx registers define the compare value itself.

The pseudo code, giving the extra cycles insertion regarding the cycle is:

```
int extra_cycle(resolution, dithercy, cycle){
  int MASK;
  int value
  switch (resolution){
    DITH4: MASK = 0x0f;
    DITH5: MASK = 0x1f;
    DITH6: MASK = 0x3f;
  }
  value = cycle * dithercy;
  if (((MASK & value) + dithercy) > MASK)
    return 1;
 return 0;
}
```

#### Dithering on Period

Writing DITHERCY in PER will lead to an average PWM period configured by the following formulas.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

DITH4 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{16} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK_TCC}x}}\right)$$

**Note:** If DITH4 mode is enabled, the last 4 significant bits from the PER, CCx, or COUNT registers correspond to the DITHERCY value, and the rest of the bits corresponds to the PER/CCx or COUNT value.

DITH5 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{32} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK_TCC}x}}\right)$$

DITH6 mode:

$$PwmPeriod = \left(\frac{\text{DITHERCY}}{64} + \text{PER}\right)\left(\frac{1}{f_{\text{GCLK_TCC}x}}\right)$$

**Dithering on Pulse-Width**

Writing DITHERCY in CCx will lead to an average PWM pulse width configured by the following formula.

DITH4 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{16} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK_TCC}x}}\right)$$

DITH5 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{32} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK_TCC}x}}\right)$$

DITH6 mode:

$$PwmPulseWidth = \left(\frac{\text{DITHERCY}}{64} + \text{CCx}\right)\left(\frac{1}{f_{\text{GCLK_TCC}x}}\right)$$

**Note:** The PWM period will remain static in this case.

### 49.6.3.4. Ramp Operations

Three ramp operation modes are supported. All of them require the timer/counter running in single-slope PWM generation. The Ramp mode is selected by writing to the Ramp Mode bits in the Waveform Control register (WAVE.RAMP).

**RAMP1 Operation**

This is the default PWM operation, described in Single-Slope PWM Generation.

**RAMP2 Operation**

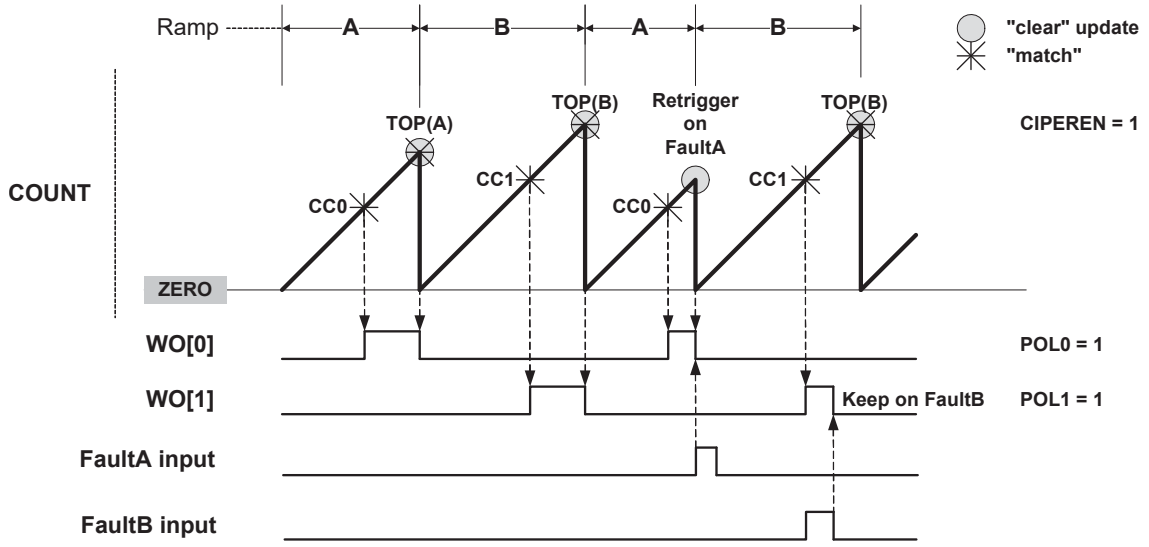
These operation modes are dedicated for power factor correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved, see Figure 49-18. In cycle A, odd channel output is disabled, and in cycle B, even channel output is disabled. The ramp index changes after each update, but can be software modified using the Ramp index command bits in Control B Set register (CTRLBSET.IDXCMD).

**Standard RAMP2 (RAMP2) Operation**

Ramp A and B periods are controlled by the PER register value. The PER value can be different on each ramp by the Circular Period buffer option in the Wave register (WAVE.CIPEREN=1). This mode uses a two-channel TCC to generate two output signals, or one output signal with another CC channel enabled in Capture mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

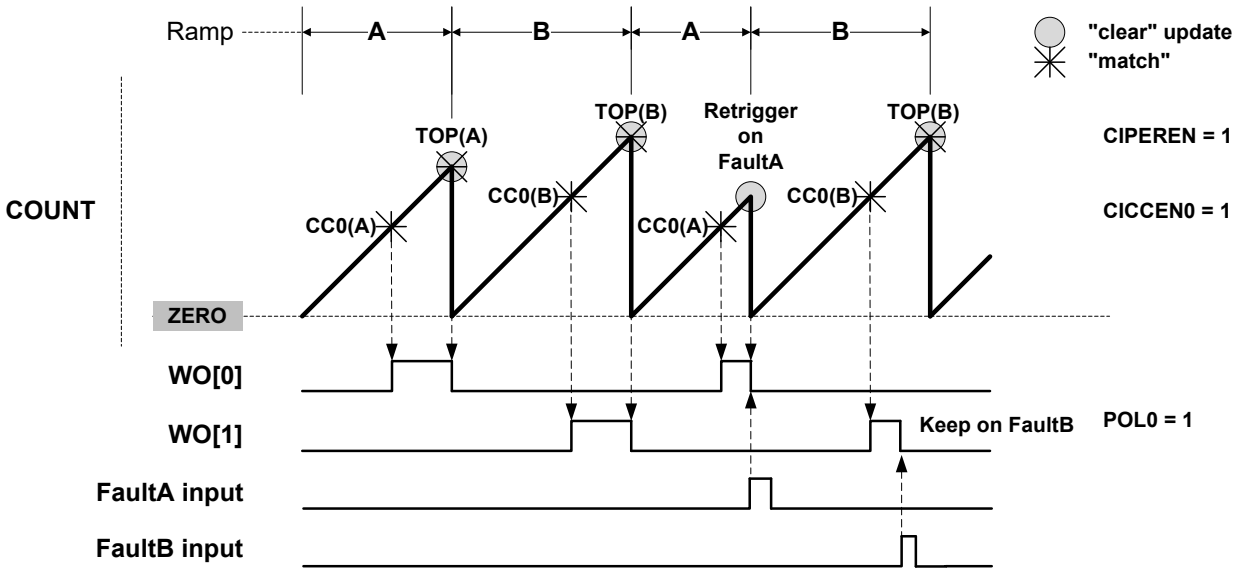
**Figure 49-18.** RAMP2 Standard Operation




## Alternate RAMP2 (RAMP2A) Operation

Alternate RAMP2 operation is similar to RAMP2, but CC0 controls both WO[0] and WO[1] waveforms when the corresponding circular buffer option is enabled (CIPEREN=1). The waveform polarity is the same on both outputs. Channel 1 can be used in capture mode.

**Figure 49-19.** RAMP2 Alternate Operation




## Critical RAMP2 (RAMP2C) Operation

Critical RAMP2 operation provides a way to cover RAMP2 operation requirements without the update constraint associated with the use of circular buffers. In this mode, CC0 is controlling the period of ramp A and PER is controlling the period of ramp B. When using more than two channels, WO[0] output is controlled by CC2 (HIGH) and CC0 (LOW). On TCC with 2 channels, a pulse on WO[0] will last the entire period of ramp A, if WAVE.POL0=0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-20.** RAMP2 Critical Operation With More Than 2 Channels

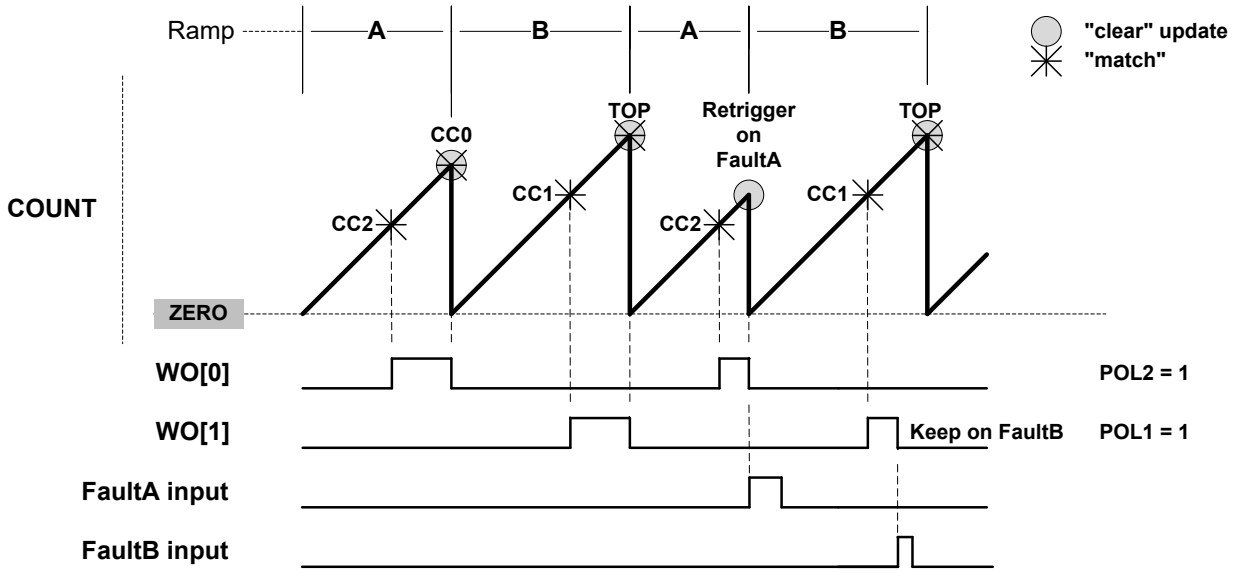
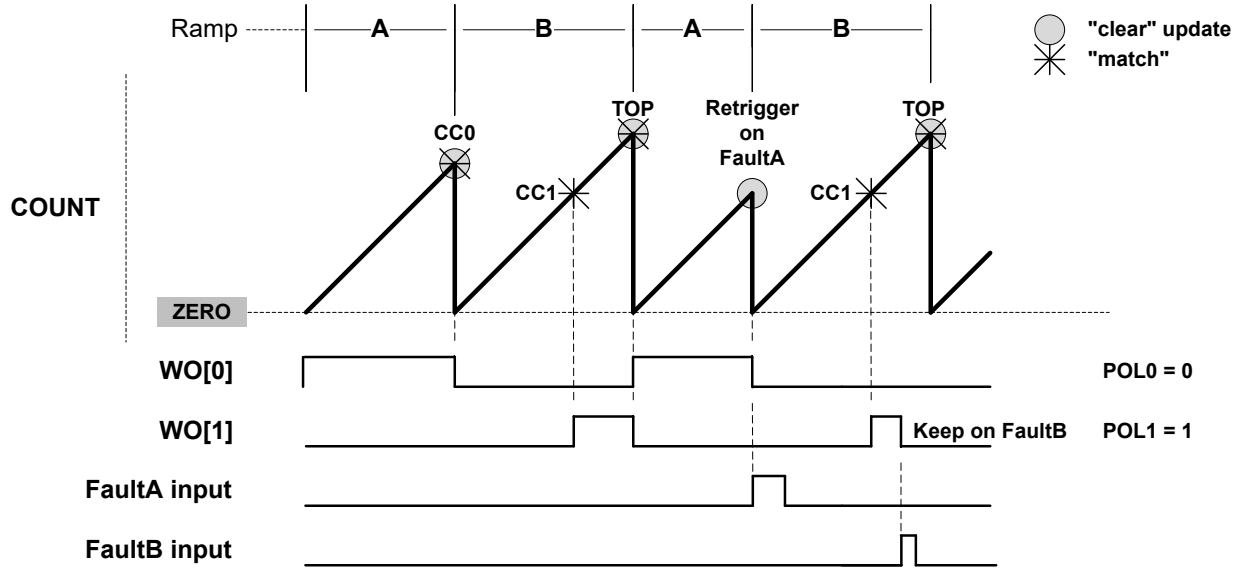



**Figure 49-21.** RAMP2 Critical Operation With 2 Channels




### 49.6.3.5. Recoverable Faults

Recoverable faults can restart or halt the timer/counter. Two faults, called Fault A and Fault B, can trigger recoverable fault actions on the compare channels CC0 and CC1 of the TCC. The compare channels' outputs can be clamped to inactive state either as long as the fault condition is present, or from the first valid fault condition detection on until the end of the timer/counter cycle.

#### Fault Inputs

The first two channel input events (TCCxMC0 and TCCxMC1) can be used as Fault A and Fault B inputs, respectively. Event system channels connected to these fault inputs must be configured as asynchronous. The TCC must work in a PWM mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## Fault Filtering

There are three filters available for each input Fault A and Fault B. They are configured by the corresponding Recoverable Fault n Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

**Input Filtering**

By default, the event detection is asynchronous. When the event occurs, the fault system will immediately and asynchronously perform the selected fault action on the compare channel output, also in device power modes where the clock is not available. To avoid false fault detection on external events (e.g. due to a glitch on an I/O port) a digital filter can be enabled and configured by the Fault B Filter Value bits in the Fault n Configuration registers (FCTRLn.FILTERVAL). If the event width is less than FILTERVAL (in clock cycles), the event will be discarded. A valid event will be delayed by FILTERVAL clock cycles.

**Fault Blanking**

This ignores any fault input for a certain time just after a selected waveform output edge. This can be used to prevent false fault triggering due to signal bouncing, as shown in the figure below. Blanking can be enabled by writing an edge triggering configuration to the Fault n Blanking Mode bits in the Recoverable Fault n Configuration register (FCTRLn.BLANK). The desired duration of the blanking must be written to the Fault n Blanking Time bits (FCTRLn.BLANKVAL).
The blanking time $t_b$ is calculated by:

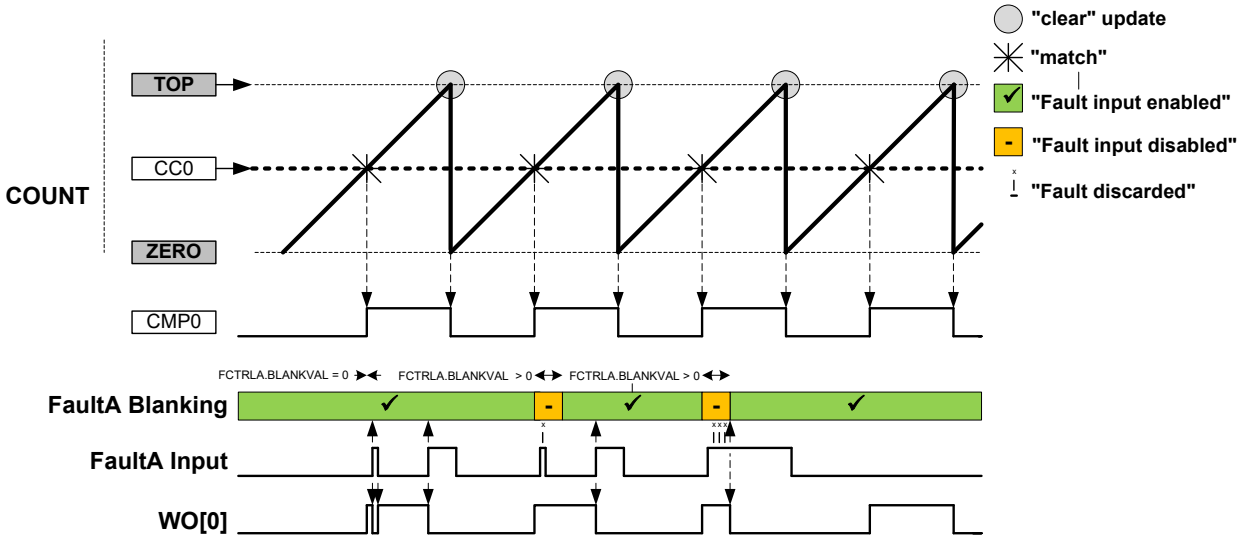
$$t_b = \frac{1 + \mathrm{BLANKVAL}}{f_{\mathrm{GCLK_TCCx_PRESC}}}$$

Where, $f_{\mathrm{GCLK_TCCx_PRESC}}$ is the frequency of the prescaled peripheral clock frequency $f_{\mathrm{GCLK_TCCx}}$.

The prescaler is enabled by writing '1' to the Fault n Blanking Prescaler bit (FCTRLn.BLANKPRESC). When disabled, $f_{\mathrm{GCLK_TCCx_PRESC}}$=$f_{\mathrm{GCLK_TCCx}}$. When enabled, $f_{\mathrm{GCLK_TCCx_PRESC}}$=$f_{\mathrm{GCLK_TCCx}}$/64.

The maximum blanking time (FCTRLn.BLANKVAL = 255) at $f_{\mathrm{GCLK_TCCx}}$ = 96 MHz is 2.67µs (no prescaler) or 170 µs (prescaling). For $f_{\mathrm{GCLK_TCCx}}$ = 1 MHz, the maximum blanking time is either 170 µs (no prescaling) or 10.9 ms (prescaling enabled).

**Figure 49-22.** Fault Blanking in RAMP1 Operation with Inverted Polarity




**Fault Qualification**

This is enabled by writing a '1' to the Fault n Qualification bit in the Recoverable Fault n Configuration register (FCTRLn.QUAL). When the recoverable fault qualification is


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

enabled (FCTRLn.QUAL = 1), the fault input is disabled all the time the corresponding channel output has an inactive level, as shown in the figures below.

**Figure 49-23.** Fault Qualification in RAMP1 Operation

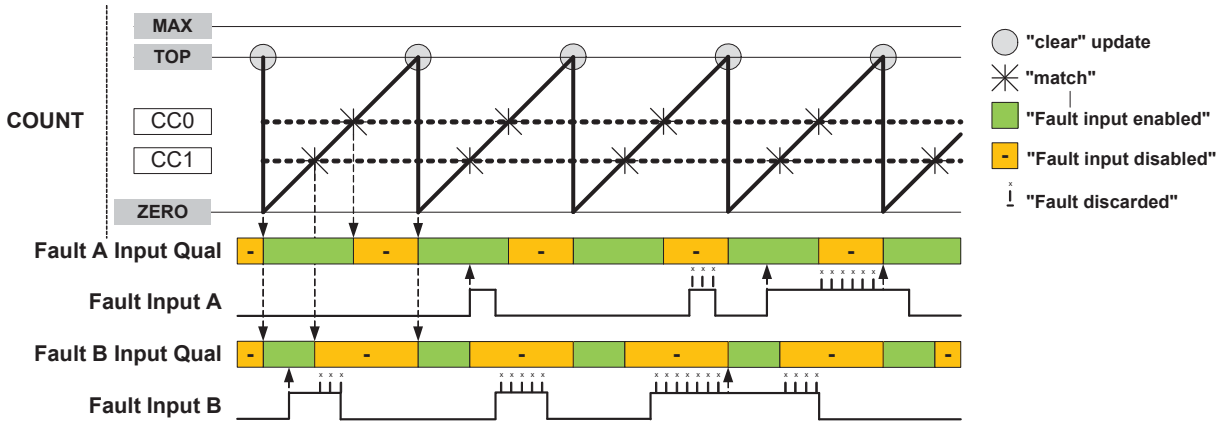
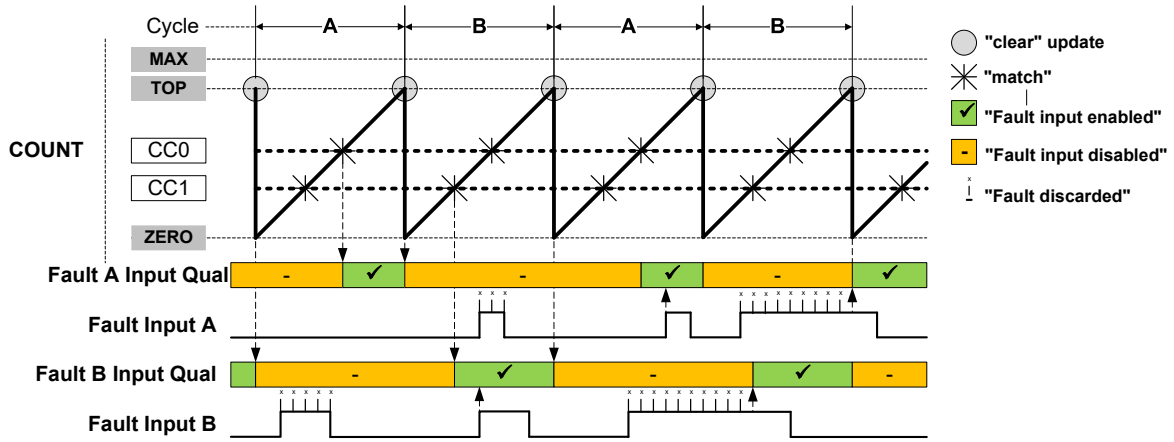



**Figure 49-24.** Fault Qualification in RAMP2 Operation with Inverted Polarity



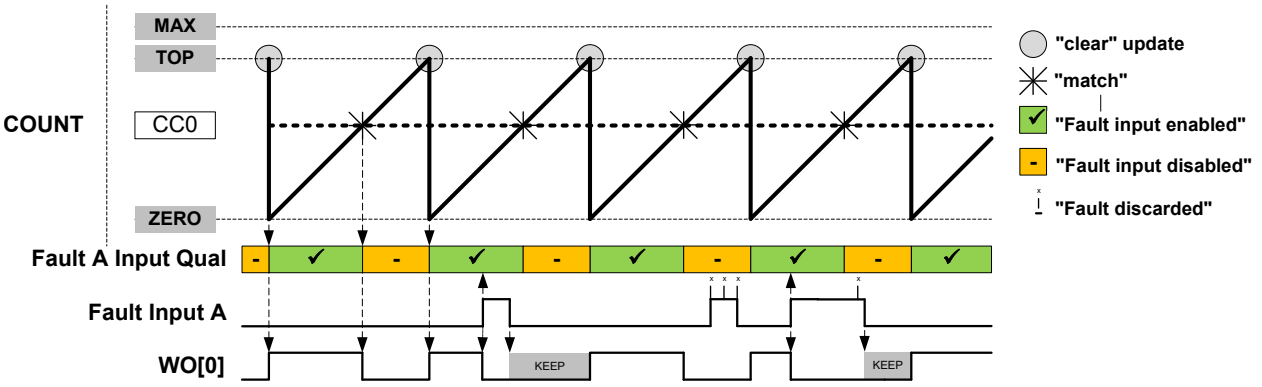

## Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; hence two or more actions can be enabled at the same time to achieve a result that is a combination of fault actions.

**Keep Action** This is enabled by writing the Fault n Keeper bit in the Recoverable Fault n Configuration register (FCTRLn.KEEP) to '1'. When enabled, the corresponding channel output will be clamped to zero as long as the fault condition is present. The clamp will be released on the start of the first cycle after the fault condition is no longer present, see next Figure.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-25.** Waveform Generation with Fault Qualification and Keep Action

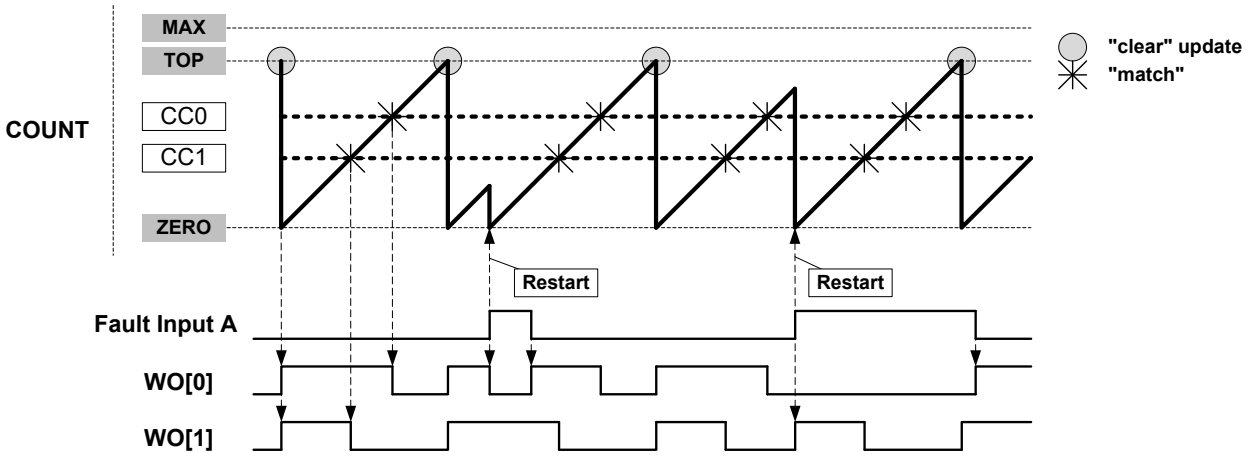



**Restart Action**

This is enabled by writing the Fault n Restart bit in Recoverable Fault n Configuration register (FCTRLn.RESTART) to '1'. When enabled, the timer/counter will be restarted as soon as the corresponding fault condition is present. The ongoing cycle is stopped and the timer/counter starts a new cycle, see Figure 49-26. In Ramp 1 mode, when the new cycle starts, the compare outputs will be clamped to inactive level as long as the fault condition is present.

**Note:** For RAMP2 operation, when a new timer/counter cycle starts the cycle index will change automatically, see Figure 49-27. Fault A and Fault B are qualified only during the cycle A and cycle B respectively: Fault A is disabled during cycle B, and Fault B is disabled during cycle A.

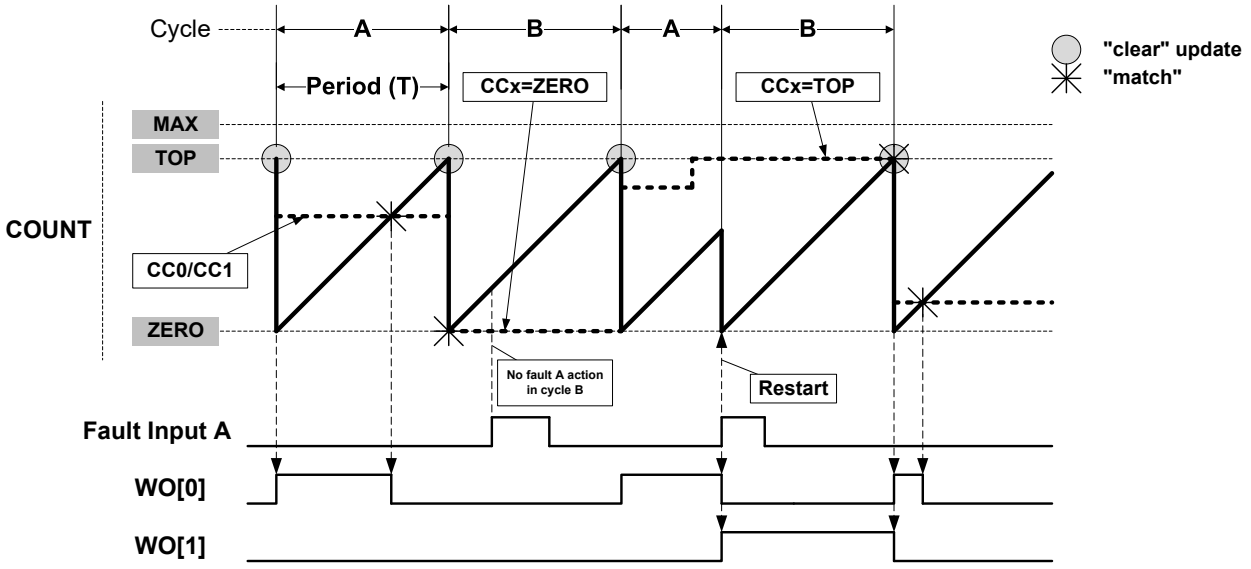
**Figure 49-26.** Waveform Generation in RAMP1 mode with Restart Action





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-27.** Waveform Generation in RAMP2 mode with Restart Action




**Capture Action**

Several capture actions can be selected by writing the Fault n Capture Action bits in the Fault n Control register (FCTRLn.CAPTURE). When one of the capture operations is selected, the counter value is captured when the fault occurs. These capture operations are available:

- CAPT – The equivalent to a standard capture operation, for additional information refer to Capture Operations.
- CAPTMIN – Gets the minimum time stamped value: on each new local minimum captured value, an event or interrupt is issued.
- CAPTMAX – Gets the maximum time stamped value: on each new local maximum captured value, an event or interrupt (IT) is issued, see Figure 49-28.
- LOCMIN – Notifies by event or interrupt when a local minimum captured value is detected.
- LOCMAX – Notifies by event or interrupt when a local maximum captured value is detected.
- DERIV0 – Notifies by event or interrupt when a local extreme captured value is detected, see Figure 49-29.

CCx Content: In CAPTMIN and CAPTMAX operations, CCx keeps the respective extremum captured values, see Figure 49-28. In LOCMIN, LOCMAX, or DERIV0 operation, CCx follows the counter value at fault time, see Figure 49-29.

Before enabling CAPTMIN or CAPTMAX mode of capture, the user must initialize the corresponding CCx register value to a value different from zero (for CAPTMIN) top (for CAPTMAX). If the CCx register initial value is zero (for CAPTMIN) top (for CAPTMAX), no captures will be performed using the corresponding channel.

MCx Behavior: In LOCMIN and LOCMAX operation, capture is performed on each capture event. The MCx interrupt flag is set only when the captured value is above or equal (for LOCMIN) or below or equal (for LOCMAX) to the previous captured value. Therefore the interrupt flag is set when a new relative local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected. DERIV0 is equivalent to an OR function of (LOCMIN, LOCMAX).

In CAPT operation, capture is performed on each capture event. The MCx interrupt flag is set on each new capture.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

In CAPTMIN and CAPTMAX operation, capture is performed only when on capture event time, the counter value is lower (for CAPTMIN) or higher (for CAPMAX) than the last captured value. The MCx interrupt flag is set only when on capture event time, the counter value is higher or equal (for CAPTMIN) or lower or equal (for CAPTMAX) to the value captured on the previous event. So interrupt flag is set when a new absolute local Minimum (for CAPTMIN) or Maximum (for CAPTMAX) value has been detected.

Interrupt Generation: In CAPT mode, an interrupt is generated on each filtered Fault n and each dedicated CCx channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

**Figure 49-28.** Capture Action “CAPTMAX”

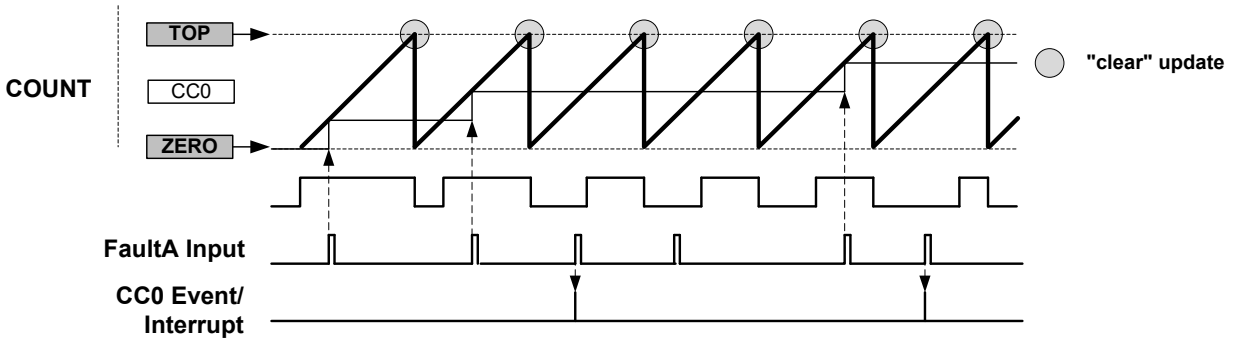
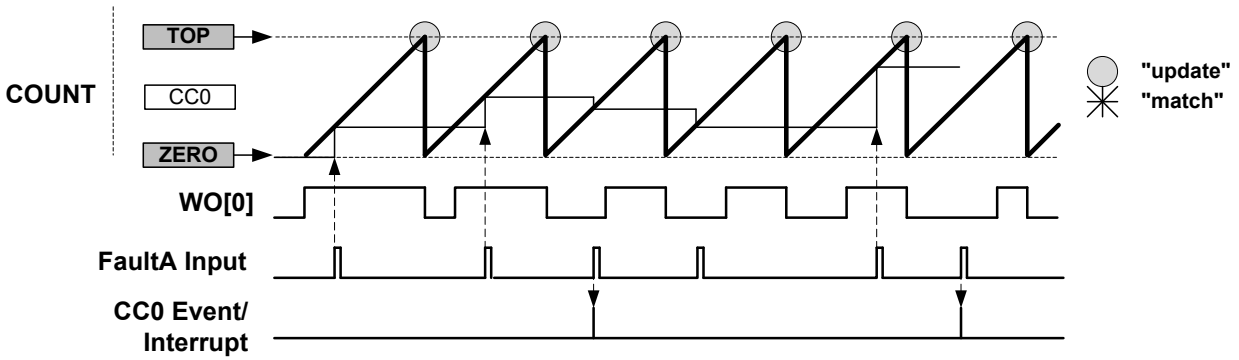



**Figure 49-29.** Capture Action “DERIV0”




**Hardware Halt Action**

This is configured by writing 0x1 to the Fault n Halt mode bits in the Recoverable Fault n Configuration register (FCTRLn.HALT). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

The next figure ('Waveform Generation with Halt and Restart Actions') shows an example where both restart action and hardware halt action are enabled for Fault A. The compare channel 0 output is clamped to inactive level as long as the timer/counter is halted. The timer/counter resumes the counting operation as soon as the fault condition is no longer present. As the restart action is enabled in this example, the timer/counter is restarted after the fault condition is no longer present.

The figure after that ('Waveform Generation with Fault Qualification, Halt, and Restart Actions') shows a similar example, but with additionally enabled fault qualification. Here, counting is resumed after the fault condition is no longer present.

Note that in RAMP2 and RAMP2A operations, when a new timer/counter cycle starts, the cycle index will automatically change.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-30.** Waveform Generation with Halt and Restart Actions

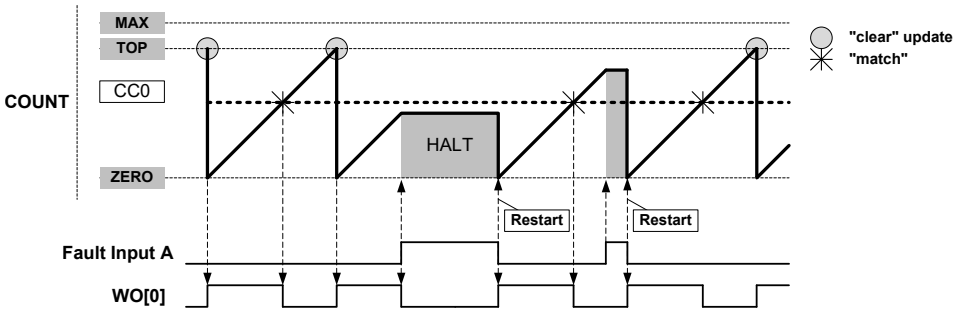
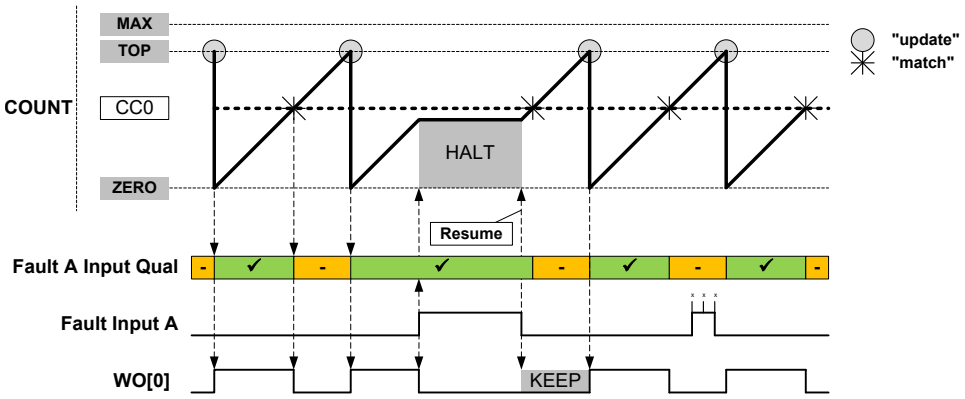
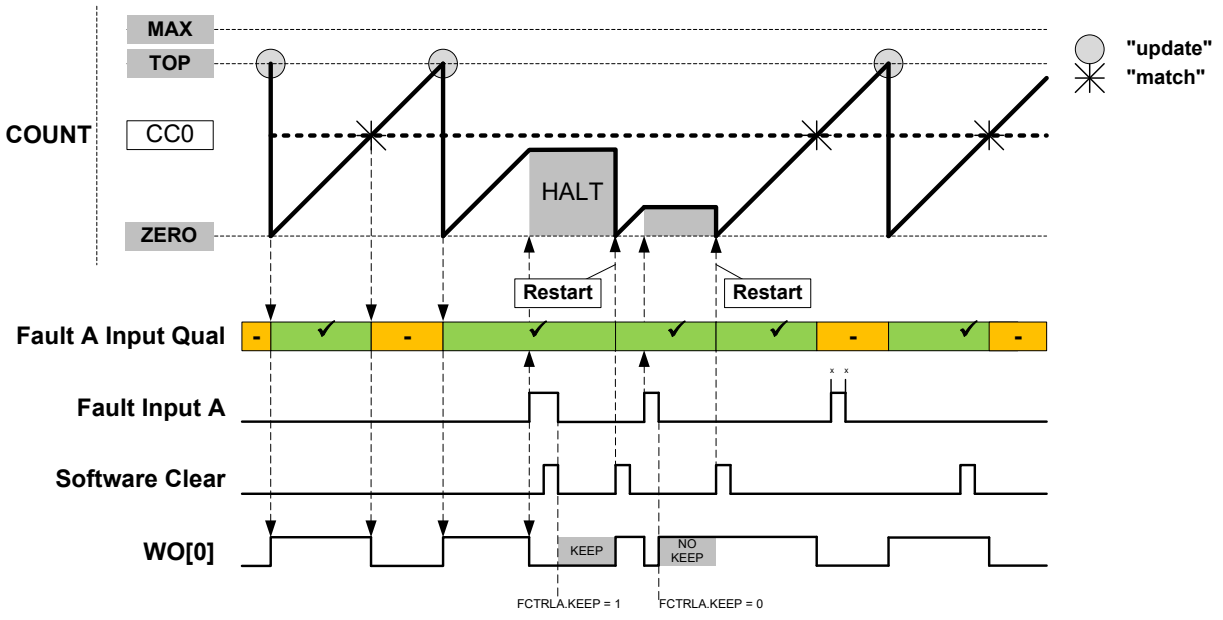



**Figure 49-31.** Waveform Generation with Fault Qualification, Halt, and Restart Actions




**Software Halt Action** This is configured by writing 0x2 to the Fault n Halt mode bits in the Recoverable Fault n configuration register (FCTRLn.HALT). Software halt action is similar to hardware halt action, but in order to restart the timer/counter, the corresponding fault condition must not be present anymore, and the corresponding FAULT n bit in the STATUS register must be cleared by software.

**Figure 49-32.** Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.6.3.6. Non-Recoverable Faults

The non-recoverable fault action will force all the compare outputs to a pre-defined level programmed into the Driver Control register (DRVCTRL.NRE and DRVCTRL.NRV). The non-recoverable fault input (EV0 and EV1) actions are enabled in Event Control register (EVCTRL.EVACT0 and EVCTRL.EVACT1).

To avoid false fault detection on external events (e.g. a glitch on an I/O port) a digital filter can be enabled using Non-Recoverable Fault Input x Filter Value bits in the Driver Control register (DRVCTRL.FILTERVALn). Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD) is written to '1', a non-recoverable Debug Faults State and an interrupt (DFS) is generated when the system goes in debug operation.

In RAMP2, RAMP2A, or DSBOTH operation, when the Lock Update bit in the Control B register is set by writing CTRLBSET.LUPD=1 and the ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

### 49.6.3.7. Time-Stamp Capture on Events or I/Os

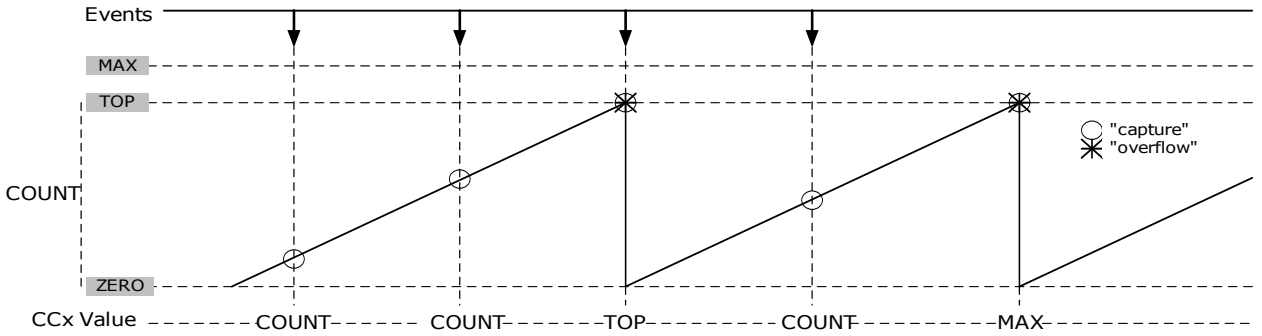
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT) is selected. The counter TOP value must be smaller than MAX.

When a capture event from the Event System or the I/O pin is detected, the COUNT value is copied into the corresponding Channel x Compare/Capture Value (CCx) register. In case of an overflow, the MAX value is copied into the corresponding CCx register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel x Interrupt Flag (INTFLAG.MCx) is set.

The timer/counter can detect capture overflow of the input capture channels: When a new capture event is detected while the Capture Channel interrupt flag (INTFLAG.MCx) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

**Figure 49-33.** Time Stamp




### 49.6.3.8. Waveform Extension

*Waveform Extension Stage Details* displays the schematic diagram of actions of the four optional units that follow the recoverable fault stage on a port pin pair: Output Matrix (OTMX), Dead-Time Insertion (DTI), SWAP and Pattern Generation. The DTI and SWAP units can be seen as a four port pair slices:

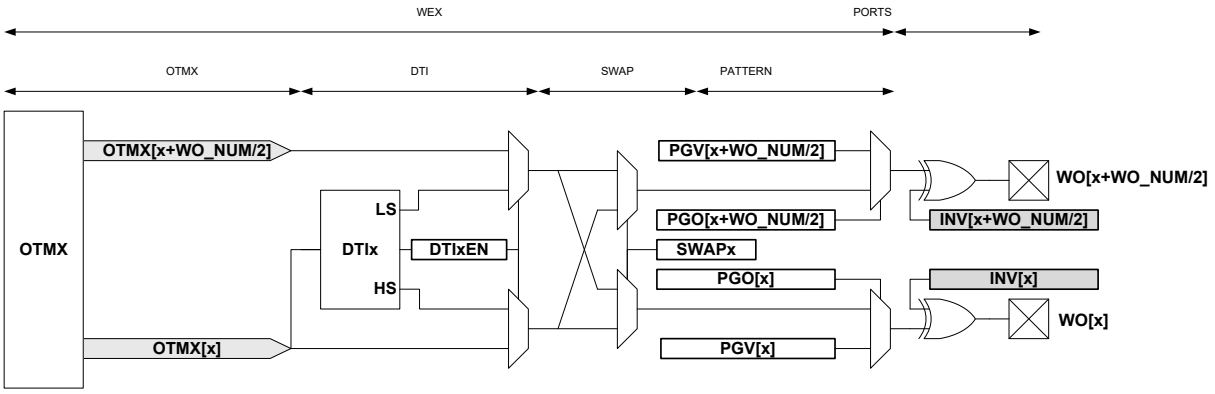
- Slice 0 DTI0 / SWAP0 acting on port pins (WO[0], WO[WO_NUM/2 +0])
- Slice 1 DTI1 / SWAP1 acting on port pins (WO[1], WO[WO_NUM/2 +1])

And generally:

- Slice n DTIx / SWAPx acting on port pins (WO[x], WO[WO_NUM/2 +x])


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-34.** Waveform Extension Stage Details




The **output matrix (OTMX)** unit distributes compare channels, according to the selectable configurations in the following table.

**Table 49-4.** Output Matrix Channel Pin Routing Configuration

| WEXCTRL.OTMX | OTMX[7] | OTMX[6] | OTMX[5] | OTMX[4] | OTMX[3] | OTMX[2] | OTMX[1] | OTMX[0] |
|---|---|---|---|---|---|---|---|---|
| 0x0 | CC1 | CC0 | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |
| 0x1 | CC1 | CC0 | CC2 | CC1 | CC0 | CC2 | CC1 | CC0 |
| 0x2 | CC0 | CC0 | CC0 | CC0 | CC0 | CC0 | CC0 | CC0 |
| 0x3 | CC1 | CC1 | CC1 | CC1 | CC1 | CC1 | CC1 | CC0 |

- Configuration 0x0 is the default configuration. The channel location is the default one and channels are distributed on outputs modulo the number of channels. Channel 0 is routed to the Output matrix output OTMX[0], and Channel 1 to OTMX[1]. If there are more outputs than channels, then channel 0 is duplicated to the Output matrix output OTMX[CC_NUM], channel 1 to OTMX[CC_NUM+1] and so on.
- Configuration 0x1 distributes the channels on output modulo half the number of channels. This assigns twice the number of output locations to the lower channels than the default configuration. This can be used, for example, to control the four transistors of a full bridge using only two compare channels.
  Using pattern generation, some of these four outputs can be overwritten by a constant level, enabling flexible drive of a full bridge in all quadrant configurations.
- Configuration 0x2 distributes compare channel 0 (CC0) to all port pins. With pattern generation, this configuration can control a stepper motor.
- Configuration 0x3 distributes the compare channel CC0 to the first output, and the channel CC1 to all other outputs. Together with pattern generation and the fault extension, this configuration can control up to seven LED strings, with a boost stage.

The table below is an example showing four compare channels on four outputs.

**Table 49-5.** Four Compare Channels on Four Outputs

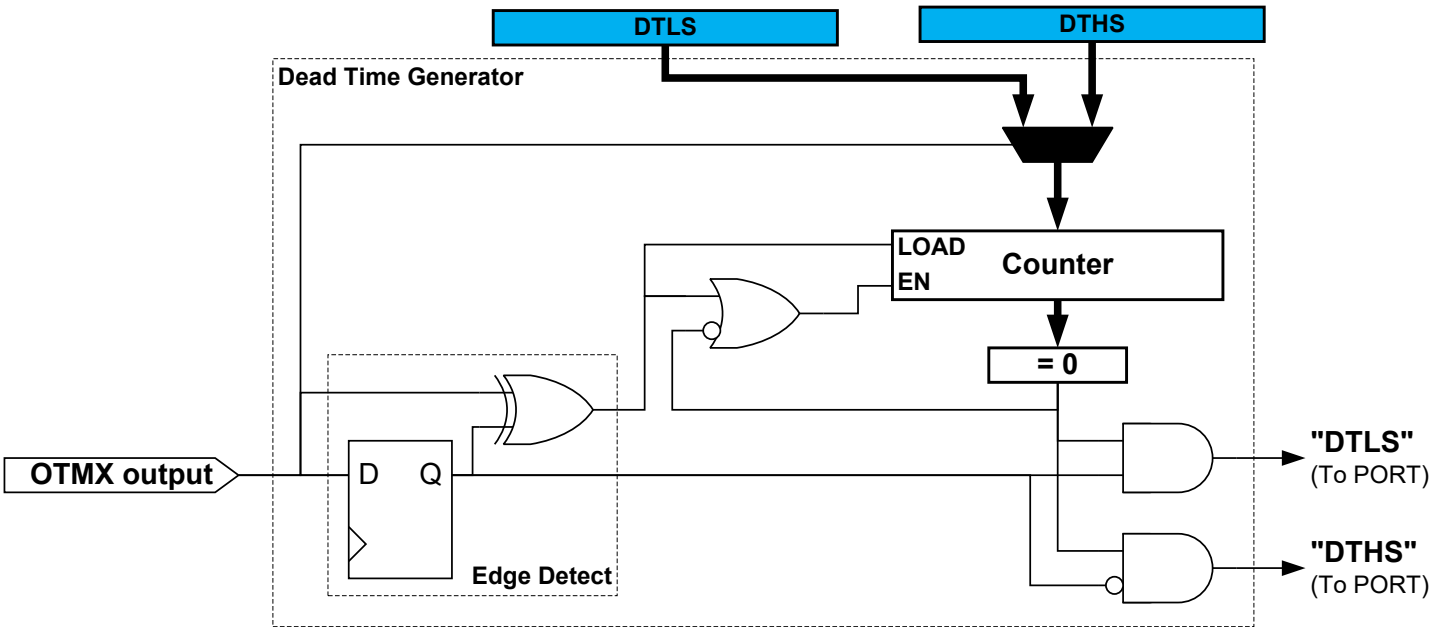
| WEXCTRL.OTMX | OTMX[3] | OTMX[2] | OTMX[1] | OTMX[0] |
|---|---|---|---|---|
| 0x0 | CC3 | CC2 | CC1 | CC0 |
| 0x1 | CC1 | CC0 | CC1 | CC0 |
| 0x2 | CC0 | CC0 | CC0 | CC0 |
| 0x3 | CC1 | CC1 | CC1 | CC0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**The dead-time insertion (DTI)** unit generates OFF time with the non-inverted low side (LS) and inverted high side (HS) of the wave generator output forced at low level. This OFF time is called dead time. Dead-time insertion ensures that the LS and HS will never switch simultaneously.

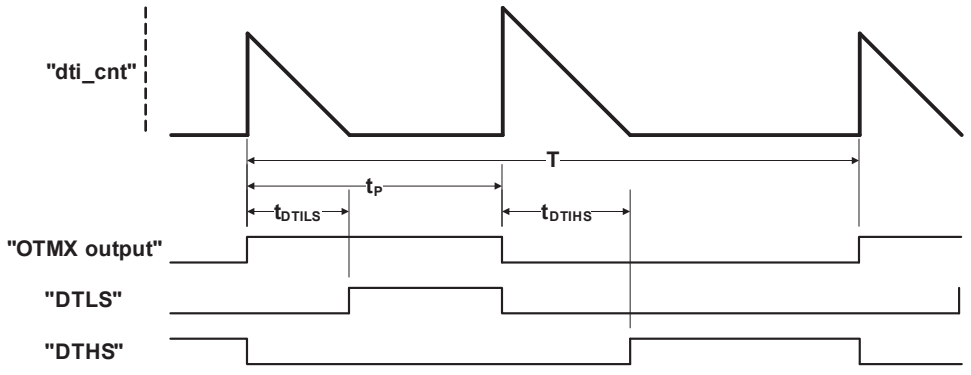
The DTI stage consists of four equal dead-time insertion generators; one for each of the first four compare channels. The following figure shows the block diagram of one DTI generator. The four channels have a common register which controls the dead time, which is independent of high side and low side setting.

**Figure 49-35.** Dead-Time Generator Block Diagram




As shown in the following figure, the 8-bit dead-time counter is decremented by one for each peripheral clock cycle until it reaches zero. A non-zero counter value will force both the low side and high side outputs into their OFF state. When the output matrix (OTMX) output changes, the dead-time counter is reloaded according to the edge of the input. When the output changes from low to high (positive edge) it initiates a counter reload of the DTLS register. When the output changes from high to low (negative edge) it reloads the DTHS register.

**Figure 49-36.** Dead-Time Generator Timing Diagram

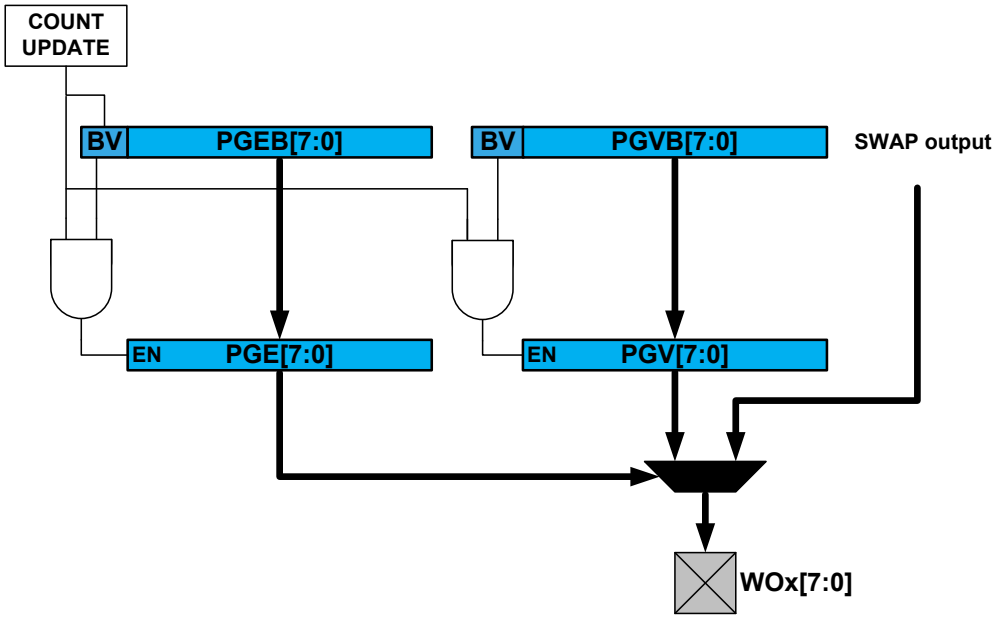



**The pattern generator unit** produces a synchronized bit pattern across the port pins it is connected to. The pattern generation features are primarily intended for handling the commutation


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

sequence in brushless DC motors (BLDC), stepper motors, and full bridge control. For more information, refer to the following figure.

**Figure 49-37.** Pattern Generator Block Diagram




As with other double-buffered timer/counter registers, the register update is synchronized to the UPDATE condition set by the timer/counter waveform generation operation. If synchronization is not required by the application, the software can simply access directly the PATT.PGE, PATT.PGV bits registers.

### 49.6.4. Host/Client Operation

Two or more TCC instances sharing the same GCLK_TCCx clock, can be linked to provide more synchronized CC channels. The operation is enabled by setting the Host Synchronization bit in Control A register (CTRLA.MSYNC) in the Client instance. When the bit is set, the Client TCC instance will synchronize the CC channels to the Host counter.

#### Related Links

CTRLA

Control A

### 49.6.5. DMA, Interrupts, and Events

**Table 49-6.** Module Requests for TCC

| Condition | Interrupt request | Event output | Event input | DMA request | DMA request is cleared |
|---|---|---|---|---|---|
| Overflow / Underflow | Yes | Yes | | Yes(1) | On DMA acknowledge |
| Channel Compare Match or Capture | Yes | Yes | Yes(2) | Yes(3) | For circular buffering: on DMA acknowledge<br>For capture channel: when CCx register is read |
| Retrigger | Yes | Yes | | | |
| Count | Yes | Yes | | | |
| Capture Overflow Error | Yes | | | | |
| Debug Fault State | Yes | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 49-6.** Module Requests for TCC (continued)

| Condition | Interrupt request | Event output | Event input | DMA request | DMA request is cleared |
|---|---|---|---|---|---|
| Recoverable Faults | Yes | | | | |
| Non-Recoverable Faults | Yes | | | | |
| TCCx Event 0 input | | | Yes(4) | | |
| TCCx Event 1 input | | | Yes(5) | | |

Notes:
1. DMA request set on Overflow, Underflow or Re-trigger conditions.
2. Can perform capture or generate recoverable fault on an event input.
3. In Capture or Circular modes.
4. On event input, either action can be executed:
   - re-trigger counter
   - control counter direction
   - stop the counter
   - decrement the counter
   - perform period and pulse width capture
   - generate non-recoverable fault
5. On event input, either action can be executed:
   - re-trigger counter
   - increment or decrement counter depending on direction
   - start the counter
   - increment or decrement counter based on direction
   - increment counter regardless of direction
   - generate non-recoverable fault

#### 49.6.5.1. DMA Operation

The TCC can generate the following DMA requests:

| | |
|---|---|
| **Counter overflow (OVF)** | If the One-shot Trigger mode in the control A register (CTRLA.DMAOS) is written to '0', the TCC generates a DMA request on each cycle when an update condition (Overflow, Underflow or Re-trigger) is detected.<br>When an update condition (Overflow, Underflow or Re-trigger) is detected while CTRLA.DMAOS=1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD=DMAOS).<br>In both cases, the request is cleared by hardware on DMA acknowledge. |
| **Channel Match (MCx)** | A DMA request is set only on a compare match if CTRLA.DMAOS=0. The request is cleared by hardware on DMA acknowledge.<br>When CTRLA.DMAOS=1, the DMA requests are not generated. |
| **Channel Capture (MCx)** | For a capture channel, the request is set when valid data is present in the CCx register, and cleared once the CCx register is read.<br>In this operation mode, the CTRLA.DMAOS bit value is ignored. |

**DMA Operation with Circular Buffer**

When circular buffer operation is enabled, the Buffer registers must be written in a correct order and synchronized to the update times of the timer. The DMA triggers of the TCC provide a way to ensure a safe and correct update of circular buffers.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

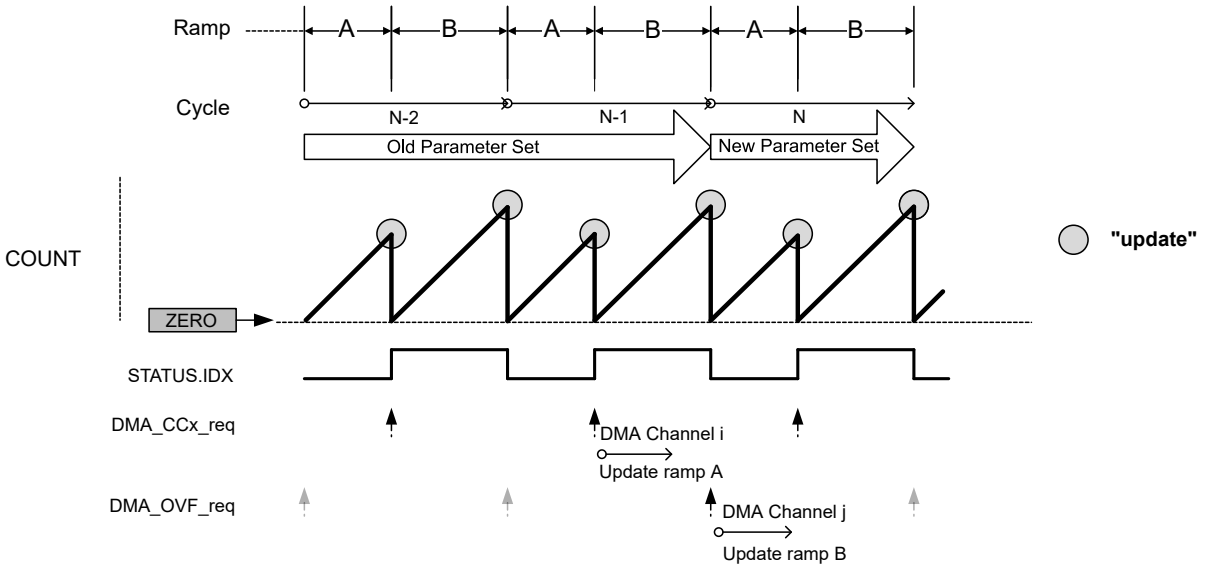
**Note:** Circular buffer are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

*DMA Operation with Circular Buffer in RAMP2 and RAMP2A Mode*
When a CCx channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of ramp B.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of ramp A with an effective DMA transfer on previous ramp B (DMA acknowledge).

The update of all circular buffer values for ramp A can be done through a DMA channel triggered on a MC trigger. The update of all circular buffer values for ramp B, can be done through a second DMA channel triggered by the overflow DMA request.

**Figure 49-38.** DMA Triggers in RAMP and RAMP2 Operation Mode and Circular Buffer Enabled



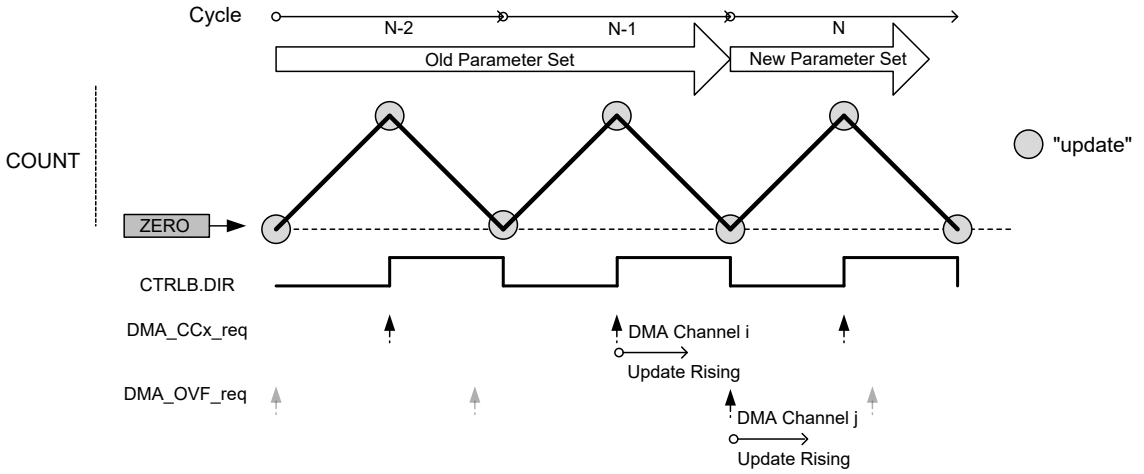

DMA Operation with Circular Buffer in DSBOTH Mode
When a CC channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of down-counting phase.

If at least one circular buffer is enabled, the DMA overflow request is conditioned to the start of up-counting phase with an effective DMA transfer on previous down-counting phase (DMA acknowledge).

When up-counting, all circular buffer values can be updated through a DMA channel triggered by MC trigger. When down-counting, all circular buffer values can be updated through a second DMA channel, triggered by the OVF DMA request.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 49-39.** DMA Triggers in DSBOTH Operation Mode and Circular Buffer Enabled




### 49.6.5.2. Interrupts

The TCC has the following interrupt sources:

- Overflow/Underflow (OVF)
- Retrigger (TRG)
- Count (CNT) - refer also to description of EVCTRL.CNTSEL.
- Capture Overflow Error (ERR)
- Non-Recoverable Update Fault (UFS)
- Debug Fault State (DFS)
- Recoverable Faults (FAULTn)
- Non-recoverable Faults (FAULTx)
- Compare Match or Capture Channels (MCx)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/ Sleep Mode Controller section for details.

Each interrupt source has an Interrupt flag associated with it. The Interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the Interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the Interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the Interrupt flag is cleared, the interrupt is disabled, or the TCC is reset. See INTFLAG for details on how to clear Interrupt flags. The TCC has one common interrupt request line for all the interrupt sources. The user must read the INTFLAG register to determine which Interrupt condition is present.

Note: Interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller

### 49.6.5.3. Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Trigger (TRG)
- Counter (CNT) For further details, refer to EVCTRL.CNTSEL description.
- Compare Match or Capture on compare/capture channels: MCx

Writing a '1' ('0') to an Event Output bit in the Event Control Register (EVCTRL.xxEO) enables (disables) the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCx):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx EV1):

- Counter re-trigger
- Counter direction control
- Stop the counter
- Decrement the counter on event
- Period and pulse width capture
- Non-recoverable fault

The TCC can take the following actions on counter Event 0 (TCCx EV0):

- Counter re-trigger
- Count on event (increment or decrement, depending on counter direction)
- Counter start - start counting on the event rising edge. Further events will not restart the counter; the counter will keep on counting using prescaled GCLK_TCCx, until it reaches TOP or ZERO, depending on the direction.
- Counter increment on event. This will increment the counter, irrespective of the counter direction.
- Count during active state of an asynchronous event (increment or decrement, depending on counter direction). In this case, the counter will be incremented or decremented on each cycle of the prescaled clock, as long as the event is active.
- Non-recoverable fault

The counter Event Actions are available in the Event Control registers (EVCTRL.EVACT0 and EVCTRL.EVACT1). For further details, refer to EVCTRL.

Writing a '1' ('0') to an Event Input bit in the Event Control register (EVCTRL.MCEIx or EVCTRL.TCEIx) enables (disables) the corresponding action on input event.

**Note:** When several events are connected to the TCC, the enabled action will apply for each of the incoming events. Refer to *EVSYS – Event System* for details on how to configure the event system.

**Related Links**

Event System (EVSYS)

### 49.6.6. Sleep Mode Operation

The TCC can be configured to operate in any Sleep mode. To be able to run in standby the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be '1'. The MODULE can in any Sleep mode wake-up the device using interrupts or perform actions through the Event System.

### 49.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Software Reset and Enable bits in Control A register (CTRLA.SWRST and CTRLA.ENABLE)

The following registers are synchronized when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Count Value register (COUNT)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

The following registers are synchronized when read:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD)
- Pattern and Pattern Buffer registers (PATT and PATTBUF)
- Waveform register (WAVE)
- Period Value and Period Buffer Value registers (PER and PERBUF)
- Compare/Capture Channel x and Channel x Compare/Capture Buffer Value registers (CCx and CCBUFx)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.

**Related Links**

Register Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 49.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | RESOLUTION[1:0] | | | | | ENABLE | SWRST |
| | | 15:8 | MSYNC | ALOCK | PRESCSYNC[1:0] | | RUNSTDBY | PRESCALER[2:0] | | |
| | | 23:16 | DMAOS | | | | | | | |
| | | 31:24 | | | CPTEN5 | CPTEN4 | CPTEN3 | CPTEN2 | CPTEN1 | CPTEN0 |
| 0x04 | CTRLBCLR | 7:0 | CMD[2:0] | | | IDXCMD[1:0] | | ONESHOT | LUPD | DIR |
| 0x05 | CTRLBSET | 7:0 | CMD[2:0] | | | IDXCMD[1:0] | | ONESHOT | LUPD | DIR |
| 0x06 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | SYNCBUSY | 7:0 | PER | WAVE | PATT | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | FCTRLA | 7:0 | RESTART | BLANK[1:0] | | QUAL | KEEP | | SRC[1:0] | |
| | | 15:8 | BLANKPRESC | CAPTURE[2:0] | | | CHSEL[1:0] | | HALT[1:0] | |
| | | 23:16 | BLANKVAL[7:0] | | | | | | | |
| | | 31:24 | | | | | FILTERVAL[3:0] | | | |
| 0x10 | FCTRLB | 7:0 | RESTART | BLANK[1:0] | | QUAL | KEEP | | SRC[1:0] | |
| | | 15:8 | BLANKPRESC | CAPTURE[2:0] | | | CHSEL[1:0] | | HALT[1:0] | |
| | | 23:16 | BLANKVAL[7:0] | | | | | | | |
| | | 31:24 | | | | | FILTERVAL[3:0] | | | |
| 0x14 | WEXCTRL | 7:0 | | | | | | | OTMX[1:0] | |
| | | 15:8 | | | | | DTIEN3 | DTIEN2 | DTIEN1 | DTIEN0 |
| | | 23:16 | DTLS[7:0] | | | | | | | |
| | | 31:24 | DTHS[7:0] | | | | | | | |
| 0x18 | DRVCTRL | 7:0 | NRE7 | NRE6 | NRE5 | NRE4 | NRE3 | NRE2 | NRE1 | NRE0 |
| | | 15:8 | NRV7 | NRV6 | NRV5 | NRV4 | NRV3 | NRV2 | NRV1 | NRV0 |
| | | 23:16 | INVEN7 | INVEN6 | INVEN5 | INVEN4 | INVEN3 | INVEN2 | INVEN1 | INVEN0 |
| | | 31:24 | FILTERVAL1[3:0] | | | | FILTERVAL0[3:0] | | | |
| 0x1C ... 0x1D | Reserved | | | | | | | | | |
| 0x1E | DBGCTRL | 7:0 | | | | | | FDDBD | | DBGRUN |
| 0x1F | Reserved | | | | | | | | | |
| 0x20 | EVCTRL | 7:0 | CNTSEL[1:0] | | EVACT1[2:0] | | | EVACT0[2:0] | | |
| | | 15:8 | TCEI1 | TCEI0 | TCINV1 | TCINV0 | | CNTEO | TRGEO | OVFEO |
| | | 23:16 | | | MCEI5 | MCEI4 | MCEI3 | MCEI2 | MCEI1 | MCEI0 |
| | | 31:24 | | | MCEO5 | MCEO4 | MCEO3 | MCEO2 | MCEO1 | MCEO0 |
| 0x24 | INTENCLR | 7:0 | | | | | ERR | CNT | TRG | OVF |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| | | 23:16 | | | MC5 | MC4 | MC3 | MC2 | MC1 | MC0 |
| | | 31:24 | | | | | | | | |
| 0x28 | INTENSET | 7:0 | | | | | ERR | CNT | TRG | OVF |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| | | 23:16 | | | MC5 | MC4 | MC3 | MC2 | MC1 | MC0 |
| | | 31:24 | | | | | | | | |
| 0x2C | INTFLAG | 7:0 | | | | | ERR | CNT | TRG | OVF |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| | | 23:16 | | | MC5 | MC4 | MC3 | MC2 | MC1 | MC0 |
| | | 31:24 | | | | | | | | |
| 0x30 | STATUS | 7:0 | PERBUFV | | PATTBUFV | SLAVE | DFS | UFS | IDX | STOP |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | FAULT1IN | FAULT0IN | FAULTBIN | FAULTAIN |
| | | 23:16 | | | CCBUFV5 | CCBUFV4 | CCBUFV3 | CCBUFV2 | CCBUFV1 | CCBUFV0 |
| | | 31:24 | | | CMP5 | CMP4 | CMP3 | CMP2 | CMP1 | CMP0 |
| 0x34 | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | COUNT[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x38 | PATT | 7:0 | PGE7 | PGE6 | PGE5 | PGE4 | PGE3 | PGE2 | PGE1 | PGE0 |
| | | 15:8 | PGV7 | PGV6 | PGV5 | PGV4 | PGV3 | PGV2 | PGV1 | PGV0 |
| 0x3A ... 0x3B | Reserved | | | | | | | | | |
| 0x3C | WAVE | 7:0 | CIPEREN | | RAMP[1:0] | | | WAVEGEN[2:0] | | |
| | | 15:8 | | | | | CICCEN3 | CICCEN2 | CICCEN1 | CICCEN0 |
| | | 23:16 | | | POL5 | POL4 | POL3 | POL2 | POL1 | POL0 |
| | | 31:24 | | | | | SWAP3 | SWAP2 | SWAP1 | SWAP0 |
| 0x40 | PER | 7:0 | PER[1:0] | | DITHER[5:0] | | | | | |
| | | 15:8 | PER[9:2] | | | | | | | |
| | | 23:16 | PER[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x44 | CC0 | 7:0 | CC[1:0] | | DITHER[5:0] | | | | | |
| | | 15:8 | CC[9:2] | | | | | | | |
| | | 23:16 | CC[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x48 | CC1 | 7:0 | CC[1:0] | | DITHER[5:0] | | | | | |
| | | 15:8 | CC[9:2] | | | | | | | |
| | | 23:16 | CC[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C | CC2 | 7:0 | CC[1:0] | | DITHER[5:0] | | | | | |
| | | 15:8 | CC[9:2] | | | | | | | |
| | | 23:16 | CC[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x50 | CC3 | 7:0 | CC[1:0] | | DITHER[5:0] | | | | | |
| | | 15:8 | CC[9:2] | | | | | | | |
| | | 23:16 | CC[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x54 | CC4 | 7:0 | CC[1:0] | | DITHER[5:0] | | | | | |
| | | 15:8 | CC[9:2] | | | | | | | |
| | | 23:16 | CC[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x58 | CC5 | 7:0 | CC[1:0] | | DITHER[5:0] | | | | | |
| | | 15:8 | CC[9:2] | | | | | | | |
| | | 23:16 | CC[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x5C ... 0x63 | Reserved | | | | | | | | | |
| 0x64 | PATTBUF | 7:0 | PGEB7 | PGEB6 | PGEB5 | PGEB4 | PGEB3 | PGEB2 | PGEB1 | PGEB0 |
| | | 15:8 | PGVB7 | PGVB6 | PGVB5 | PGVB4 | PGVB3 | PGVB2 | PGVB1 | PGVB0 |
| 0x66 ... 0x6B | Reserved | | | | | | | | | |
| 0x6C | PERBUF | 7:0 | PERBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| | | 15:8 | PERBUF[9:2] | | | | | | | |
| | | 23:16 | PERBUF[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x70 | CCBUF0 | 7:0 | CCBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| | | 15:8 | CCBUF[9:2] | | | | | | | |
| | | 23:16 | CCBUF[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x74 | CCBUF1 | 7:0 | CCBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| | | 15:8 | CCBUF[9:2] | | | | | | | |
| | | 23:16 | CCBUF[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x78 | CCBUF2 | 7:0 | CCBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| | | 15:8 | CCBUF[9:2] | | | | | | | |
| | | 23:16 | CCBUF[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x7C | CCBUF3 | 7:0 | CCBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| | | 15:8 | CCBUF[9:2] | | | | | | | |
| | | 23:16 | CCBUF[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x80 | CCBUF4 | 7:0 | CCBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| | | 15:8 | CCBUF[9:2] | | | | | | | |
| | | 23:16 | CCBUF[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x84 | CCBUF5 | 7:0 | CCBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| | | 15:8 | CCBUF[9:2] | | | | | | | |
| | | 23:16 | CCBUF[17:10] | | | | | | | |
| | | 31:24 | | | | | | | | |

## 49.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized (ENABLE, SWRST)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | CPTEN5 | CPTEN4 | CPTEN3 | CPTEN2 | CPTEN1 | CPTEN0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DMAOS | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MSYNC | ALOCK | PRESCSYNC[1:0] | | RUNSTDBY | PRESCALER[2:0] | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | RESOLUTION[1:0] | | | | | ENABLE | SWRST |
| Access | | R/W | R/W | | | | R/W | R/W |
| Reset | | 0 | 0 | | | | 0 | 0 |

**Bits 24, 25, 26, 27, 28, 29 – CPTENx** Capture Channel x Enable
These bits are used to select the capture or compare operation on channel x.
Writing a '1' to CPTENx enables capture on channel x.
Writing a '0' to CPTENx disables capture on channel x.

**Bit 23 – DMAOS** DMA One-Shot Trigger Mode
This bit enables the DMA One-shot Trigger Mode.
Writing a '1' to this bit will generate a DMA trigger on TCC cycle following a TCC_CTRLBSET_CMD_DMAOS command.
Writing a '0' to this bit will generate DMA triggers on each TCC cycle.
This bit is not synchronized.

**Bit 15 – MSYNC** Host Synchronization (only for TCC Client instance)
This bit must be set if the TCC counting operation must be synchronized on its Host TCC.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The TCC controls its own counter. |
| 1 | The counter is controlled by its Host TCC. |

**Bit 14 – ALOCK** Auto Lock
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The Lock Update bit in the Control B register (CTRLB.LUPD) is not affected by overflow/underflow, and re-trigger events |
| 1 | CTRLB.LUPD is set to '1' on each overflow/underflow or re-trigger event. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 13:12 – PRESCSYNC[1:0]** Prescaler and Counter Synchronization

These bits select if on re-trigger event, the Counter is cleared or reloaded on either the next GCLK_TCCx clock, or on the next prescaled GCLK_TCCx clock. It is also possible to reset the prescaler on re-trigger event.
These bits are not synchronized.

| Value | Name | Description | |
|---|---|---|---|
| | | Counter Reloaded | Prescaler |
| 0x0 | GCLK | Reload or reset Counter on next GCLK | - |
| 0x1 | PRESC | Reload or reset Counter on next prescaler clock | - |
| 0x2 | RESYNC | Reload or reset Counter on next GCLK | Reset prescaler counter |
| 0x3 | Reserved | | |

**Bit 11 – RUNSTDBY** Run in Standby

This bit is used to keep the TCC running in Standby mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The TCC is halted in standby. |
| 1 | The TCC continues to run in standby. |

**Bits 10:8 – PRESCALER[2:0]** Prescaler

These bits select the Counter prescaler factor.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV1 | Prescaler: GCLK_TCCx |
| 0x1 | DIV2 | Prescaler: GCLK_TCCx/2 |
| 0x2 | DIV4 | Prescaler: GCLK_TCCx/4 |
| 0x3 | DIV8 | Prescaler: GCLK_TCCx/8 |
| 0x4 | DIV16 | Prescaler: GCLK_TCCx/16 |
| 0x5 | DIV64 | Prescaler: GCLK_TCCx/64 |
| 0x6 | DIV256 | Prescaler: GCLK_TCCx/256 |
| 0x7 | DIV1024 | Prescaler: GCLK_TCCx/1024 |

**Bits 6:5 – RESOLUTION[1:0]** Dithering Resolution

These bits increase the TCC resolution by enabling the dithering options.
These bits are not synchronized.

**Table 49-7.** Dithering

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | The dithering is disabled. |
| 0x1 | DITH4 | Dithering is done every 16 PWM frames. PER[3:0] and CCx[3:0] contain dithering pattern selection. |
| 0x2 | DITH5 | Dithering is done every 32 PWM frames. PER[4:0] and CCx[4:0] contain dithering pattern selection. |
| 0x3 | DITH6 | Dithering is done every 64 PWM frames. PER[5:0] and CCx[5:0] contain dithering pattern selection. |

**Bit 1 – ENABLE** Enable

Due to synchronization there is delay from writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately and the ENABLE bit in the SYNCBUSY register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the TCC (except DBGCTRL) to their initial state, and the TCC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.

**Notes:**

1. When the CTRLA.SWRST is written, the user must poll the SYNCBUSY.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers or bits without SWRST are disallowed until the SYNCBUSY.SWRST bit is cleared by hardware.

| Value | Description |
|---|---|
| 0 | There is no Reset operation ongoing. |
| 1 | The Reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.2. Control B Clear

**Name:** CTRLBCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | IDXCMD[1:0] | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** TCC Command
Writing a '0' to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

**Bits 4:3 – IDXCMD[1:0]** Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.
Writing zero to these bits has no effect.
Writing a '1' to any of these bits will clear the pending command.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | DISABLE Command disabled: IDX toggles between cycles A and B |
| 0x1 | SET | Set IDX: cycle B will be forced in the next cycle |
| 0x2 | CLEAR | Clear IDX: cycle A will be forced in next cycle |
| 0x3 | HOLD | Hold IDX: the next cycle will be the same as the current cycle. |

**Bit 2 – ONESHOT** One-Shot
This bit controls one-shot operation of the TCC. When one-shot operation is enabled, the TCC will stop counting on the next overflow/underflow condition or on a stop command.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will disable the one-shot operation.

| Value | Description |
|---|---|
| 0 | The TCC will update the counter value on overflow/underflow condition and continue operation. |
| 1 | The TCC will stop counting on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the TCC buffered registers.
When CTRLB.LUPD is cleared, the hardware UPDATE registers with value from their buffered registers is enabled.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the registers updates on hardware UPDATE condition.

| Value | Description |
|---|---|
| 0 | The CCBx, PERB, PGVB, and PGEB buffer registers values *are* copied into the corresponding CCx, PER, PGV, and PGE registers on hardware update condition. |
| 1 | The CCBx, PERB, PGVB, and PGEB buffer registers values are *not* copied into the corresponding CCx, PER, PGV, and PGE registers on hardware update condition. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 0 – DIR** Counter Direction

This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will clear the bit and make the counter count up.

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.3. Control B Set

**Name:** CTRLBSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | IDXCMD[1:0] | | ONESHOT | LUPD | DIR |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:5 – CMD[2:0]** TCC Command
These bits can be used for software control of re-triggering and stop commands of the TCC. When a command has been executed, the CMD bit field will be read back as zero. The commands are executed on the next prescaled GCLK_TCCx clock cycle.
Writing zero to this bit group has no effect
Writing a value different from 0x0 to this bit field will issue a command for execution.

**Important:** This command requires synchronization before being executed.
A valid sequence is the following:

- Issue CMD command (CTRLBSET.CMD = command)
- Wait for CMD synchronization (SYNCBUSY.CTRLB = 0)
- Wait for CMD read back as zero (CTRLBSET.CMD = 0)

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | RETRIGGER | Force start, restart or retrigger |
| 0x2 | STOP | Force stop |
| 0x3 | UPDATE | Force update of double buffered registers |
| 0x4 | READSYNC | Force a read synchronization of COUNT |
| 0x5 | DMAOS | One-shot DMA trigger |

**Bits 4:3 – IDXCMD[1:0]** Ramp Index Command
These bits can be used to force cycle A and cycle B changes in RAMP2 and RAMP2A operation. On timer/counter update condition, the command is executed, the IDX flag in STATUS register is updated and the IDXCMD command is cleared.
Writing a zero to these bits has no effect.
Writing a valid value to these bits will set a command.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Command disabled: IDX toggles between cycles A and B |
| 0x1 | SET | Set IDX: cycle B will be forced in the next cycle |
| 0x2 | CLEAR | Clear IDX: cycle A will be forced in next cycle |
| 0x3 | HOLD | Hold IDX: the next cycle will be the same as the current cycle. |

**Bit 2 – ONESHOT** One-Shot
This bit controls one-shot operation of the TCC. When in one-shot operation, the TCC will stop counting on the next overflow/underflow condition or a stop command.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the one-shot operation.

| Value | Description |
|---|---|
| 0 | The TCC will count continuously. |
| 1 | The TCC will stop counting on the next underflow/overflow condition. |

**Bit 1 – LUPD** Lock Update

This bit controls the update operation of the TCC buffered registers.
When CTRLB.LUPD is set, the hardware UPDATE registers with value from their buffered registers is disabled. Disabling the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
This bit has no effect when input capture operation is enabled.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will disable the registers updates on hardware UPDATE condition.

| Value | Description |
|---|---|
| 0 | The CCBx, PERB, PGVB, and PGEB buffer registers values *are* copied into the corresponding CCx, PER, PGV, and PGE registers on hardware update condition. |
| 1 | The CCBx, PERB, PGVB, and PGEB buffer registers values are *not* copied into CCx, PER, PGV, and PGE registers on hardware update condition. |

**Bit 0 – DIR** Counter Direction

This bit is used to change the direction of the counter.
Writing a '0' to this bit has no effect
Writing a '1' to this bit will set the bit and make the counter count down.
**Note:** When the TCC is counting down, the COUNT register must be initialized to the TOP value (PER or CC0 value depending on the mode).

| Value | Description |
|---|---|
| 0 | The timer/counter is counting up (incrementing). |
| 1 | The timer/counter is counting down (decrementing). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.4. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | CC5 | CC4 | CC3 | CC2 | CC1 | CC0 |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER | WAVE | PATT | COUNT | STATUS | CTRLB | ENABLE | SWRST |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 8, 9, 10, 11, 12, 13 – CC** Compare/Capture Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare/Capture Channel x register between the clock domains is complete.
This bit is set when the synchronization of Compare/Capture Channel x register between clock domains is started.
CCx bit is available only for existing Compare/Capture Channels. For details on CC channels number, refer to each TCC feature list.
This bit is set when the synchronization of CCx register between clock domains is started.

**Bit 7 – PER** PER Synchronization Busy
This bit is cleared when the synchronization of PER register between the clock domains is complete.
This bit is set when the synchronization of PER register between clock domains is started.

**Bit 6 – WAVE** WAVE Synchronization Busy
This bit is cleared when the synchronization of WAVE register between the clock domains is complete.
This bit is set when the synchronization of WAVE register between clock domains is started.

**Bit 5 – PATT** PATT Synchronization Busy
This bit is cleared when the synchronization of PATTERN register between the clock domains is complete.
This bit is set when the synchronization of PATTERN register between clock domains is started.

**Bit 4 – COUNT** COUNT Synchronization Busy
This bit is cleared when the synchronization of COUNT register between the clock domains is complete.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit is set when the synchronization of COUNT register between clock domains is started.

**Bit 3 – STATUS** STATUS Synchronization Busy
This bit is cleared when the synchronization of STATUS register between the clock domains is complete.
This bit is set when the synchronization of STATUS register between clock domains is started.

**Bit 2 – CTRLB** CTRLB Synchronization Busy
This bit is cleared when the synchronization of CTRLB register between the clock domains is complete.
This bit is set when the synchronization of CTRLB register between clock domains is started.

**Bit 1 – ENABLE** ENABLE Synchronization Busy
This bit is cleared when the synchronization of ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of ENABLE bit between clock domains is started.

**Bit 0 – SWRST** SWRST Synchronization Busy
This bit is cleared when the synchronization of SWRST bit between the clock domains is complete.
This bit is set when the synchronization of SWRST bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.5. Fault Control A and B

**Name:** FCTRLn
**Offset:** 0x0C + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | FILTERVAL[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | BLANKVAL[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BLANKPRESC | CAPTURE[2:0] | | | CHSEL[1:0] | | HALT[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RESTART | BLANK[1:0] | | QUAL | KEEP | | SRC[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | | 0 | 0 |

**Bits 27:24 – FILTERVAL[3:0]** Recoverable Fault n Filter Value
These bits define the filter value applied on MCEx (x=0,1) event input line. The value must be set to zero when MCEx event is used as synchronous event.

**Bits 23:16 – BLANKVAL[7:0]** Recoverable Fault n Blanking Value
These bits determine the duration of the blanking of the fault input source. Activation and edge selection of the blank filtering are done by the BLANK bits (FCTRLn.BLANK).
When enabled, the fault input source is internally disabled for BLANKVAL* prescaled GCLK_TCCx periods after the detection of the waveform edge.

**Bit 15 – BLANKPRESC** Recoverable Fault n Blanking Value Prescaler
This bit enables a factor 64 prescaler factor on used as base frequency of the BLANKVAL value.

| Value | Description |
|---|---|
| 0 | Blank time is BLANKVAL* prescaled GCLK_TCCx. |
| 1 | Blank time is BLANKVAL* 64 * prescaled GCLK_TCCx. |

**Bits 14:12 – CAPTURE[2:0]** Recoverable Fault n Capture Action
These bits select the capture and Fault n interrupt/event conditions.

**Table 49-8.** Fault n Capture Action

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Capture on valid recoverable Fault n is disabled |
| 0x1 | CAPT | On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each new captured value. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 49-8.** Fault n Capture Action (continued)

| Value | Name | Description |
|---|---|---|
| 0x2 | CAPTMIN | On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local minimum detection. |
| 0x3 | CAPTMAX | On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0], if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTn flag rises on each local maximun detection. |
| 0x4 | LOCMIN | On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local minimum value detection. |
| 0x5 | LOCMAX | On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun detection. |
| 0x6 | DERIV0 | On rising edge of a valid recoverable Fault n, capture counter value on channel selected by CHSEL[1:0]. INTFLAG.FAULTn flag rises on each local maximun or minimum detection. |
| 0x7 | CAPTMARK | Capture with ramp index as MSB value. |

**Bits 11:10 – CHSEL[1:0]** Recoverable Fault n Capture Channel
These bits select the channel for capture operation triggered by recoverable Fault n.

| Value | Name | Description |
|---|---|---|
| 0x0 | CC0 | Capture value stored into CC0 |
| 0x1 | CC1 | Capture value stored into CC1 |
| 0x2 | CC2 | Capture value stored into CC2 |
| 0x3 | CC3 | Capture value stored into CC3 |

**Bits 9:8 – HALT[1:0]** Recoverable Fault n Halt Operation
These bits select the halt action for recoverable Fault n.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Halt action disabled |
| 0x1 | HW | Hardware halt action |
| 0x2 | SW | Software halt action |
| 0x3 | NR | Non-recoverable fault |

**Bit 7 – RESTART** Recoverable Fault n Restart
Setting this bit enables restart action for Fault n.

| Value | Description |
|---|---|
| 0 | Fault n restart action is disabled. |
| 1 | Fault n restart action is enabled. |

**Bits 6:5 – BLANK[1:0]** Recoverable Fault n Blanking Operation
These bits, select the blanking start point for recoverable Fault n.

| Value | Name | Description |
|---|---|---|
| 0x0 | START | Blanking applied from start of the Ramp period |
| 0x1 | RISE | Blanking applied from rising edge of the waveform output |
| 0x2 | FALL | Blanking applied from falling edge of the waveform output |
| 0x3 | BOTH | Blanking applied from each toggle of the waveform output |

**Bit 4 – QUAL** Recoverable Fault n Qualification
Setting this bit enables the recoverable Fault n input qualification.

| Value | Description |
|---|---|
| 0 | The recoverable Fault n input is not disabled on CMPx value condition. |
| 1 | The recoverable Fault n input is disabled when output signal is at inactive level (CMPx == 0). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 3 – KEEP** Recoverable Fault n Keep

Setting this bit enables the Fault n keep action.

| Value | Description |
|---|---|
| 0 | The Fault n state is released as soon as the recoverable Fault n is released. |
| 1 | The Fault n state is released at the end of TCC cycle. |

**Bits 1:0 – SRC[1:0]** Recoverable Fault n Source

These bits select the TCC event input for recoverable Fault n.
Event system channel connected to MCEx event input, must be configured to route the event asynchronously, when used as a recoverable Fault n input.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Fault input disabled |
| 0x1 | ENABLE | MCEx (x=0,1) event input |
| 0x2 | INVERT | Inverted MCEx (x=0,1) event input |
| 0x3 | ALTFAULT | Alternate fault (A or B) state at the end of the previous period. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.6. Waveform Extension Control

**Name:** WEXCTRL
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DTHS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DTLS[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DTIEN3 | DTIEN2 | DTIEN1 | DTIEN0 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | OTMX[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 31:24 – DTHS[7:0]** Dead-Time High Side Outputs Value
This register holds the number of GCLK_TCCx clock cycles for the dead-time high side.

**Bits 23:16 – DTLS[7:0]** Dead-time Low Side Outputs Value
This register holds the number of GCLK_TCCx clock cycles for the dead-time low side.

**Bits 8, 9, 10, 11 – DTIENx** Dead-time Insertion Generator x Enable
Setting any of these bits enables the dead-time insertion generator for the corresponding output matrix. This will override the output matrix [x] and [x+WO_NUM/2], with the low side and high side waveform respectively.

| Value | Description |
|---|---|
| 0 | No dead-time insertion override. |
| 1 | Dead time insertion override on signal outputs[x] and [x+WO_NUM/2], from matrix outputs[x] signal. |

**Bits 1:0 – OTMX[1:0]** Output Matrix
These bits define the matrix routing of the TCC waveform generation outputs to the port pins, according to Waveform Extension.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.7. Driver Control

**Name:** DRVCTRL
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | FILTERVAL1[3:0] | | | | FILTERVAL0[3:0] | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | INVEN7 | INVEN6 | INVEN5 | INVEN4 | INVEN3 | INVEN2 | INVEN1 | INVEN0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | NRV7 | NRV6 | NRV5 | NRV4 | NRV3 | NRV2 | NRV1 | NRV0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | NRE7 | NRE6 | NRE5 | NRE4 | NRE3 | NRE2 | NRE1 | NRE0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:28 – FILTERVAL1[3:0]** Non-Recoverable Fault Input 1 Filter Value
These bits define the filter value applied on TCE1 event input line. When the TCE1 event input line is configured as a synchronous event, this value must be 0x0.

**Bits 27:24 – FILTERVAL0[3:0]** Non-Recoverable Fault Input 0 Filter Value
These bits define the filter value applied on TCE0 event input line. When the TCE0 event input line is configured as a synchronous event, this value must be 0x0.

**Bits 16, 17, 18, 19, 20, 21, 22, 23 – INVEN** Waveform Output x Inversion
These bits are used to select inversion on the output of channel x.
Writing a '1' to INVENx inverts output from WO[x].
Writing a '0' to INVENx disables inversion of output from WO[x].

**Bits 8, 9, 10, 11, 12, 13, 14, 15 – NRV** NRVx Non-Recoverable State x Output Value
These bits define the value of the enabled override outputs, under non-recoverable fault condition.

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – NRE** Non-Recoverable State x Output Enable
These bits enable the override of individual outputs by NRVx value, under non-recoverable fault condition.

| Value | Description |
|---|---|
| 0 | Non-recoverable fault tri-state the output. |
| 1 | Non-recoverable faults set the output to NRVx level. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.8. Debug control

**Name:** DBGCTRL
**Offset:** 0x1E
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | FDDBD | | DBGRUN |
| Access | | | | | | R/W | | R/W |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – FDDBD** Fault Detection on Debug Break Detection

This bit is not affected by software Reset and should not be changed by software while the TCC is enabled.

By default this bit is zero, and the on-chip debug (OCD) fault protection is disabled. When this bit is written to '1', OCD break request from the OCD system will trigger non-recoverable fault. When this bit is set, OCD fault protection is enabled and OCD break request from the OCD system will trigger a non-recoverable fault.

| Value | Description |
|---|---|
| 0 | No faults are generated when TCC is halted in Debug mode. |
| 1 | A non recoverable fault is generated and FAULTD flag is set when TCC is halted in Debug mode. |

**Bit 0 – DBGRUN** Debug Running State

This bit is not affected by software Reset and should not be changed by software while the TCC is enabled.

| Value | Description |
|---|---|
| 0 | The TCC is halted when the device is halted in Debug mode. |
| 1 | The TCC continues normal operation when the device is halted in Debug mode. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.9. Event Control

**Name:** EVCTRL
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | MCEO5 | MCEO4 | MCEO3 | MCEO2 | MCEO1 | MCEO0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | MCEI5 | MCEI4 | MCEI3 | MCEI2 | MCEI1 | MCEI0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TCEI1 | TCEI0 | TCINV1 | TCINV0 | | CNTEO | TRGEO | OVFEO |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CNTSEL[1:0] | | EVACT1[2:0] | | | EVACT0[2:0] | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 24, 25, 26, 27, 28, 29 – MCEO** Match or Capture Channel x Event Output Enable
These bits control if the match/capture event on channel x is enabled and will be generated for every match or capture.

| Value | Description |
|---|---|
| 0 | Match/capture x event is disabled and will not be generated. |
| 1 | Match/capture x event is enabled and will be generated for every compare/capture on channel x. |

**Bits 16, 17, 18, 19, 20, 21 – MCEI** Match or Capture Channel x Event Input Enable
These bits indicate if the match/capture x incoming event is enabled
These bits are used to enable match or capture input events to the CCx channel of TCC.

| Value | Description |
|---|---|
| 0 | Incoming events are disabled. |
| 1 | Incoming events are enabled. |

**Bits 14, 15 – TCEI** Timer/Counter Event Input x Enable
This bit is used to enable input event x to the TCC.

| Value | Description |
|---|---|
| 0 | Incoming event x is disabled. |
| 1 | Incoming event x is enabled. |

**Bits 12, 13 – TCINV** Timer/Counter Event x Invert Enable
This bit inverts the event x input.

| Value | Description |
|---|---|
| 0 | Input event source x is not inverted. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Input event source x is inverted. |

**Bit 10 – CNTEO** Timer/Counter Event Output Enable

This bit is used to enable the counter cycle event. When enabled, an event will be generated on begin or end of counter cycle depending of CNTSEL[1:0] settings.

| Value | Description |
|---|---|
| 0 | Counter cycle output event is disabled and will not be generated. |
| 1 | Counter cycle output event is enabled and will be generated depend of CNTSEL[1:0] value. |

**Bit 9 – TRGEO** Retrigger Event Output Enable

This bit is used to enable the counter retrigger event. When enabled, an event will be generated when the counter retriggers operation.

| Value | Description |
|---|---|
| 0 | Counter retrigger event is disabled and will not be generated. |
| 1 | Counter retrigger event is enabled and will be generated for every counter retrigger. |

**Bit 8 – OVFEO** Overflow/Underflow Event Output Enable

This bit is used to enable the overflow/underflow event. When enabled an event will be generated when the counter reaches the TOP or the ZERO value.

| Value | Description |
|---|---|
| 0 | Overflow/underflow counter event is disabled and will not be generated. |
| 1 | Overflow/underflow counter event is enabled and will be generated for every counter overflow/underflow. |

**Bits 7:6 – CNTSEL[1:0]** Timer/Counter Interrupt and Event Output Selection

These bits define on which part of the counter cycle the counter event output is generated.

| Value | Name | Description |
|---|---|---|
| 0x0 | BEGIN | An interrupt/event is generated at begin of each counter cycle |
| 0x1 | END | An interrupt/event is generated at end of each counter cycle |
| 0x2 | BETWEEN | An interrupt/event is generated between each counter cycle. |
| 0x3 | BOUNDARY | An interrupt/event is generated at begin of first counter cycle, and end of last counter cycle. |

**Bits 5:3 – EVACT1[2:0]** Timer/Counter Event Input 1 Action

These bits define the action the TCC will perform on TCE1 event input.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Event action disabled. |
| 0x1 | RETRIGGER | Start, restart or re-trigger TC on event |
| 0x2 | DIR (asynch) | Direction control |
| 0x3 | STOP | Stop TC on event |
| 0x4 | DEC | Decrement TC on event |
| 0x5 | PPW | Period captured into CC0 Pulse Width on CC1 |
| 0x6 | PWP | Period captured into CC1 Pulse Width on CC0 |
| 0x7 | FAULT | Non-recoverable Fault |

**Bits 2:0 – EVACT0[2:0]** Timer/Counter Event Input 0 Action

These bits define the action the TCC will perform on TCE0 event input 0.

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Event action disabled. |
| 0x1 | RETRIGGER | Start, restart or re-trigger TC on event |
| 0x2 | COUNTEV | Count on event. |
| 0x3 | START | Start TC on event |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0x4 | INC | Increment TC on EVENT |
| 0x5 | COUNT (async) | Count on active state of asynchronous event |
| 0x6 | STAMP | Capture overflow times (Max value) |
| 0x7 | FAULT | Non-recoverable Fault |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 49.8.10. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | MC5 | MC4 | MC3 | MC2 | MC1 | MC0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ERR | CNT | TRG | OVF |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 16, 17, 18, 19, 20, 21 – MCx** Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which disables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 15 – FAULT1** Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault x interrupt is disabled. |
| 1 | The Non-Recoverable Fault x interrupt is enabled. |

**Bit 14 – FAULT0** Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault x interrupt is disabled. |
| 1 | The Non-Recoverable Fault x interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 13 – FAULTB** Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Recoverable Fault B Interrupt Disable/Enable bit, which disables the Recoverable Fault B interrupt.

| Value | Description |
|---|---|
| 0 | The Recoverable Fault B interrupt is disabled. |
| 1 | The Recoverable Fault B interrupt is enabled. |

**Bit 12 – FAULTA** Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Recoverable Fault A Interrupt Disable/Enable bit, which disables the Recoverable Fault A interrupt.

| Value | Description |
|---|---|
| 0 | The Recoverable Fault A interrupt is disabled. |
| 1 | The Recoverable Fault A interrupt is enabled. |

**Bit 11 – DFS** Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Debug Fault State Interrupt Disable/Enable bit, which disables the Debug Fault State interrupt.

| Value | Description |
|---|---|
| 0 | The Debug Fault State interrupt is disabled. |
| 1 | The Debug Fault State interrupt is enabled. |

**Bit 10 – UFS** Non-Recoverable Update Fault Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will clear the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which disables the Non-Recoverable Update Fault interrupt.
**Note:** This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Update Fault interrupt is disabled. |
| 1 | The Non-Recoverable Update Fault interrupt is enabled. |

**Bit 3 – ERR** Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Compare interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 2 – CNT** Counter Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Counter Interrupt Disable/Enable bit, which disables the Counter interrupt.

| Value | Description |
|---|---|
| 0 | The Counter interrupt is disabled. |
| 1 | The Counter interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 1 – TRG** Retrigger Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Retrigger Interrupt Disable/Enable bit, which disables the Retrigger interrupt.

| Value | Description |
|---|---|
| 0 | The Retrigger interrupt is disabled. |
| 1 | The Retrigger interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 49.8.11. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** PAC Write-Protection

This register allows the user to enable an interrupt without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | MC5 | MC4 | MC3 | MC2 | MC1 | MC0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ERR | CNT | TRG | OVF |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 16, 17, 18, 19, 20, 21 – MCx** Match or Capture Channel x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the corresponding Match or Capture Channel x Interrupt Disable/Enable bit, which enables the Match or Capture Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match or Capture Channel x interrupt is disabled. |
| 1 | The Match or Capture Channel x interrupt is enabled. |

**Bit 15 – FAULT1** Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Non-Recoverable Fault x Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault x interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault x interrupt is disabled. |
| 1 | The Non-Recoverable Fault x interrupt is enabled. |

**Bit 14 – FAULT0** Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault x interrupt is disabled. |
| 1 | The Non-Recoverable Fault x interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 13 – FAULTB** Recoverable Fault B Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Recoverable Fault B Interrupt Disable/Enable bit, which enables the Recoverable Fault B interrupt.

| Value | Description |
|---|---|
| 0 | The Recoverable Fault B interrupt is disabled. |
| 1 | The Recoverable Fault B interrupt is enabled. |

**Bit 12 – FAULTA** Recoverable Fault A Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Recoverable Fault A Interrupt Disable/Enable bit, which enables the Recoverable Fault A interrupt.

| Value | Description |
|---|---|
| 0 | The Recoverable Fault A interrupt is disabled. |
| 1 | The Recoverable Fault A interrupt is enabled. |

**Bit 11 – DFS** Non-Recoverable Debug Fault Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Debug Fault State Interrupt Disable/Enable bit, which enables the Debug Fault State interrupt.

| Value | Description |
|---|---|
| 0 | The Debug Fault State interrupt is disabled. |
| 1 | The Debug Fault State interrupt is enabled. |

**Bit 10 – UFS** Non-Recoverable Update Fault Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit will set the Non-Recoverable Update Fault Interrupt Disable/Enable bit, which enables the Non-Recoverable Update Fault interrupt.
**Note:** This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Update Fault interrupt is disabled. |
| 1 | The Non-Recoverable Update Fault interrupt is enabled. |

**Bit 3 – ERR** Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Compare interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 2 – CNT** Counter Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Counter interrupt.

| Value | Description |
|---|---|
| 0 | The Counter interrupt is disabled. |
| 1 | The Counter interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 1 – TRG** Retrigger Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Retrigger Interrupt Disable/Enable bit, which enables the Retrigger interrupt.

| Value | Description |
|---|---|
| 0 | The Retrigger interrupt is disabled. |
| 1 | The Retrigger interrupt is enabled. |

**Bit 0 – OVF** Overflow Interrupt Enable

Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Overflow Interrupt Disable/Enable bit, which enables the Overflow interrupt request.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 49.8.12. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x2C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | MC5 | MC4 | MC3 | MC2 | MC1 | MC0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ERR | CNT | TRG | OVF |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 16, 17, 18, 19, 20, 21 – MCx** Match or Capture Channel x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a match with the compare condition or once CCx register contain a valid capture value.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match or Capture Channel x interrupt flag
In Capture operation, this flag is automatically cleared when CCx register is read.

**Bit 15 – FAULT1** Non-Recoverable Fault x Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Non-Recoverable Fault x occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault x interrupt flag.

**Bit 14 – FAULT0** Non-Recoverable Fault x Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault x Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault x interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault x interrupt is disabled. |
| 1 | The Non-Recoverable Fault x interrupt is enabled. |

**Bit 13 – FAULTB** Recoverable Fault B Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 12 – FAULTA** Recoverable Fault A Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a Recoverable Fault B occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Recoverable Fault B interrupt flag.

**Bit 11 – DFS** Non-Recoverable Debug Fault State Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an Debug Fault State occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Debug Fault State interrupt flag.

**Bit 10 – UFS** Non-Recoverable Update Fault
This flag is set when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD).
Writing a zero to this bit has no effect.
Writing a one to this bit clears the Non-Recoverable Update Fault interrupt flag.
**Note:** This bit is only available on variant L devices. Refer to the *Configuration Summary* for more information.

**Bit 3 – ERR** Error Interrupt Flag
This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel x interrupt flag is one. In which case there is nowhere to store the new capture.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the error interrupt flag.

**Bit 2 – CNT** Counter Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter event occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the CNT interrupt flag.

**Bit 1 – TRG** Retrigger Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after a counter retrigger occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the re-trigger interrupt flag.

**Bit 0 – OVF** Overflow Interrupt Flag
This flag is set on the next CLK_TCC_COUNT cycle after an overflow condition occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.13. Status

**Name:** STATUS
**Offset:** 0x30
**Reset:** 0x00000001
**Property:** Read-Synchronized, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | CMP5 | CMP4 | CMP3 | CMP2 | CMP1 | CMP0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | CCBUFV5 | CCBUFV4 | CCBUFV3 | CCBUFV2 | CCBUFV1 | CCBUFV0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FAULT1 | FAULT0 | FAULTB | FAULTA | FAULT1IN | FAULT0IN | FAULTBIN | FAULTAIN |
| Access | R/W | R/W | R/W | R/W | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PERBUFV | | PATTBUFV | SLAVE | DFS | UFS | IDX | STOP |
| Access | R/W | | R/W | R | R/W | R/W | R | R |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 1 |

**Bits 24, 25, 26, 27, 28, 29 – CMPx** Channel x Compare Value
This bit reflects the channel x output compare value.

| Value | Description |
|---|---|
| 0 | Channel compare output value is 0. |
| 1 | Channel compare output value is 1. |

**Bits 16, 17, 18, 19, 20, 21 – CCBUFVx** Channel x Compare or Capture Buffer Valid
For a compare channel, this bit is set when a new value is written to the corresponding CCBUFx register. The bit is cleared either by writing a '1' to the corresponding location when CTRLB.LUPD is set, or automatically on an UPDATE condition.
For a capture channel, the bit is set when a valid capture value is stored in the CCBUFx register. The bit is automatically cleared when the CCx register is read.

**Bits 14, 15 – FAULTx** Non-recoverable Fault x State
This bit is set by hardware as soon as non-recoverable Fault x condition occurs.
This bit is cleared by writing a one to this bit and when the corresponding FAULTxIN status bit is low. Once this bit is clear, the timer/counter will restart from the last COUNT value. To restart the timer/counter from BOTTOM, the timer/counter restart command must be executed before clearing the corresponding STATEx bit. For further details on timer/counter commands, refer to available commands description (CTRLBSET.CMD).

**Bit 13 – FAULTB** Recoverable Fault B State
This bit is set by hardware as soon as recoverable Fault B condition occurs.
This bit can be clear by hardware when Fault B action is resumed, or by writing a '1' to this bit when the corresponding FAULTBIN bit is low. If software halt command is enabled (FAULTB.HALT=SW), clearing this bit will release the timer/counter.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 12 – FAULTA** Recoverable Fault A State
This bit is set by hardware as soon as recoverable Fault A condition occurs.
This bit can be clear by hardware when Fault A action is resumed, or by writing a '1' to this bit when the corresponding FAULTAIN bit is low. If software halt command is enabled (FAULTA.HALT=SW), clearing this bit will release the timer/counter.

**Bit 11 – FAULT1IN** Non-Recoverable Fault 1 Input
This bit is set while an active Non-Recoverable Fault 1 input is present.

**Bit 10 – FAULT0IN** Non-Recoverable Fault 0 Input
This bit is set while an active Non-Recoverable Fault 0 input is present.

**Bit 9 – FAULTBIN** Recoverable Fault B Input
This bit is set while an active Recoverable Fault B input is present.

**Bit 8 – FAULTAIN** Recoverable Fault A Input
This bit is set while an active Recoverable Fault A input is present.

**Bit 7 – PERBUFV** Period Buffer Valid
This bit is set when a new value is written to the PERBUF register. This bit is automatically cleared by hardware on UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit.

**Bit 5 – PATTBUFV** Pattern Generator Value Buffer Valid
This bit is set when a new value is written to the PATTBUF register. This bit is automatically cleared by hardware on UPDATE condition when CTRLB.LUPD is set, or by writing a '1' to this bit.

**Bit 4 – SLAVE** Client
This bit is set when TCC is set in Client mode. This bit follows the CTRLA.MSYNC bit state.

**Bit 3 – DFS** Debug Fault State
This bit is set by hardware in Debug mode when DDBGCTRL.FDDBD bit is set. The bit is cleared by writing a '1' to this bit and when the TCC is not in Debug mode.
When the bit is set, the counter is halted and the Waveforms state depend on DRVCTRL.NRE and DRVCTRL.NRV registers.

**Bit 2 – UFS** Non-recoverable Update Fault State
This bit is set by hardware when the RAMP index changes and the Lock Update bit is set (CTRLBSET.LUPD). The bit is cleared by writing a one to this bit.
When the bit is set, the waveforms state depend on DRVCTRL.NRE and DRVCTRL.NRV registers.

**Bit 1 – IDX** Ramp Index
In RAMP2 and RAMP2A operation, the bit is cleared during the cycle A and set during the cycle B. In RAMP1 operation, the bit always reads zero. For details on ramp operations, refer to Ramp Operations.

**Bit 0 – STOP** Stop
This bit is set when the TCC is disabled either on a STOP command or on an UPDATE condition when One-Shot operation mode is enabled (CTRLBSET.ONESHOT=1).
This bit is clear on the next incoming counter increment or decrement.

| Value | Description |
|---|---|
| 0 | Counter is running. |
| 1 | Counter is stopped. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.14. Counter Value

**Name:** COUNT
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

**Note:** Prior to any read access, this register must be synchronized by user by writing the according TCC Command value to the Control B Set register (CTRLBSET.CMD = READSYNC).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:0 – COUNT[23:0]** Counter Value
These bits hold the value of the Counter register.
**Note:** When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

**Note:** This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [23:m] |
|---|---|
| 0x0 - NONE | 23:0 (depicted) |
| 0x1 - DITH4 | 23:4 |
| 0x2 - DITH5 | 23:5 |
| 0x3 - DITH6 | 23:6 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.15. Pattern

**Name:** PATT
**Offset:** 0x38
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PGV7 | PGV6 | PGV5 | PGV4 | PGV3 | PGV2 | PGV1 | PGV0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PGE7 | PGE6 | PGE5 | PGE4 | PGE3 | PGE2 | PGE1 | PGE0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 8, 9, 10, 11, 12, 13, 14, 15 – PGV** Pattern Generation Output Value
This register holds the values of pattern for each waveform output.

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PGE** Pattern Generation Output Enable
This register holds the enable status of pattern generation for each waveform output. A bit written to '1' will override the corresponding SWAP output with the corresponding PGVn value.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.16. Waveform

**Name:** WAVE
**Offset:** 0x3C
**Reset:** 0x00000000
**Property:** Write-Synchronized Bits, Enable-Protected (WAVEGEN, RAMP)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SWAP3 | SWAP2 | SWAP1 | SWAP0 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | POL5 | POL4 | POL3 | POL2 | POL1 | POL0 |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CICCEN3 | CICCEN2 | CICCEN1 | CICCEN0 |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CIPEREN | | RAMP[1:0] | | | WAVEGEN[2:0] | | |
| Access | R/W | | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | | 0 | 0 | | 0 | 0 | 0 |

**Bits 24, 25, 26, 27 – SWAP** Swap DTI Output Pair x
Setting these bits enables output swap of DTI outputs [x] and [x+WO_NUM/2]. Note the DTIxEN settings will not affect the swap operation.

**Bits 16, 17, 18, 19, 20, 21 – POL** Channel Polarity x
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.

| Value | Name | Description |
|---|---|---|
| 0 | (single-slope PWM waveform generation) | Compare output is initialized to ~DIR and set to DIR when TCC counter matches CCx value |
| 1 | (single-slope PWM waveform generation) | Compare output is initialized to DIR and set to ~DIR when TCC counter matches CCx value. |
| 0 | (dual-slope PWM waveform generation) | Compare output is set to ~DIR when TCC counter matches CCx value |
| 1 | (dual-slope PWM waveform generation) | Compare output is set to DIR when TCC counter matches CCx value. |

**Bits 8, 9, 10, 11 – CICCEN** Circular CC Enable x
Setting this bits enables the compare circular buffer option on the first four Compare/Capture channels. When the bit is set, CCx register value is copied-back into the CCx register on UPDATE condition.

**Bit 7 – CIPEREN** Circular Period Enable
Setting this bits enable the period circular buffer option. When the bit is set, the PER register value is copied-back into the PERB register on UPDATE condition.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 5:4 – RAMP[1:0]** Ramp Operation

These bits select Ramp operation (RAMP). These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | RAMP1 | RAMP1 operation |
| 0x1 | RAMP2A | Alternative RAMP2 operation |
| 0x2 | RAMP2 | RAMP2 operation |
| 0x3 | RAMP2C | Critical RAMP2 operation |
| 0x4 | - | Reserved |

**Bits 2:0 – WAVEGEN[2:0]** Waveform Generation Operation

These bits select the waveform generation operation. The settings impact the top value and control if frequency or PWM waveform generation should be used. These bits are not synchronized.

| Value | Name | Description | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Operation | Top | Update | Waveform Output On Match | Waveform Output On Update | OVFIF/Event Up | Down |
| 0x0 | NFRQ | Normal Frequency | PER | TOP/Zero | Toggle | Stable | TOP | Zero |
| 0x1 | MFRQ | Match Frequency | CC0 | TOP/Zero | Toggle | Stable | TOP | Zero |
| 0x2 | NPWM | Normal PWM | PER | TOP/Zero | Set | Clear | TOP | Zero |
| 0x3 | Reserved | - | - | - | - | - | - | - |
| 0x4 | DSCRITICAL | Dual-slope PWM | PER | Zero | ~DIR | Stable | – | Zero |
| 0x5 | DSBOTTOM | Dual-slope PWM | PER | Zero | ~DIR | Stable | – | Zero |
| 0x6 | DSBOTH | Dual-slope PWM | PER | TOP & Zero | ~DIR | Stable | TOP | Zero |
| 0x7 | DSTOP | Dual-slope PWM | PER | Zero | ~DIR | Stable | TOP | – |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.17. Period Value

**Name:** PER
**Offset:** 0x40
**Reset:** 0xFFFFFFFF
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PER[17:10] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PER[9:2] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PER[1:0] | | DITHER[5:0] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 23:6 – PER[17:0]** Period Value

These bits hold the value of the TCC period count.
**Note:** When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

**Note:** This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [23:m] |
|---|---|
| 0x0 - NONE | 23:0 |
| 0x1 - DITH4 | 23:4 |
| 0x2 - DITH5 | 23:5 |
| 0x3 - DITH6 | 23:6 (depicted) |

**Bits 5:0 – DITHER[5:0]** Dithering Cycle Number

These bits hold the number of extra cycles that are added on the PWM pulse period every 64 PWM frames.
**Note:** This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [n:0] |
|---|---|
| 0x0 - NONE | - |
| 0x1 - DITH4 | 3:0 |
| 0x2 - DITH5 | 4:0 |
| 0x3 - DITH6 | 5:0 (depicted) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.18. Compare/Capture Channel x

**Name:** CCx
**Offset:** 0x44 + x*0x04 [x=0..5]
**Reset:** 0x00000000
**Property:** Write-Synchronized

The CCx register represents the 16-, 24- bit value, CCx. The register has two functions, depending of the mode of operation.

For capture operation, this register represents the second buffer level and access point for the CPU and DMA.

For compare operation, this register is continuously compared to the counter value. Normally, the output form the comparator is then used for generating waveforms.

CCx register is updated with the buffer value from their corresponding CCBUFx register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CC[17:10] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CC[9:2] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC[1:0] | | DITHER[5:0] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:6 – CC[17:0]** Channel x Compare/Capture Value
These bits hold the value of the Channel x compare/capture register.
**Notes:**

1. When the TCC is configured as a 16-bit timer/counter, the excess bits are read as zero.
2. This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [23:m] |
|---|---|
| 0x0 - NONE | 23:0 |
| 0x1 - DITH4 | 23:4 |
| 0x2 - DITH5 | 23:5 |
| 0x3 - DITH6 | 23:6 (depicted) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 5:0 – DITHER[5:0]** Dithering Cycle Number

These bits hold the number of extra cycles that are added on the PWM pulse width every 64 PWM frames.

**Note:** This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [n:0] |
|---|---|
| 0x0 - NONE | - |
| 0x1 - DITH4 | 3:0 |
| 0x2 - DITH5 | 4:0 |
| 0x3 - DITH6 | 5:0 (depicted) |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 49.8.19. Pattern Buffer

**Name:** PATTBUF
**Offset:** 0x64
**Reset:** 0x0000
**Property:** Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PGVB7 | PGVB6 | PGVB5 | PGVB4 | PGVB3 | PGVB2 | PGVB1 | PGVB0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PGEB7 | PGEB6 | PGEB5 | PGEB4 | PGEB3 | PGEB2 | PGEB1 | PGEB0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 8, 9, 10, 11, 12, 13, 14, 15 – PGVB** Pattern Generation Output Value Buffer
This register is the buffer for the PGV register. If double buffering is used, valid content in this register is copied to the PGV register on an UPDATE condition.

**Bits 0, 1, 2, 3, 4, 5, 6, 7 – PGEB** Pattern Generation Output Enable Buffer
This register is the buffer of the PGE register. If double buffering is used, valid content in this register is copied into the PGE register at an UPDATE condition.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.20. Period Buffer Value

**Name:** PERBUF
**Offset:** 0x6C
**Reset:** 0xFFFFFFFF
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PERBUF[17:10] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PERBUF[9:2] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PERBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 23:6 – PERBUF[17:0]** Period Buffer Value
These bits hold the value of the Period Buffer register. The value is copied to PER register on UPDATE condition.
**Note:** When the TCC is configured as 16-bit timer/counter, the excess bits are read zero.

**Note:** This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [23:m] |
|---|---|
| 0x0 - NONE | 23:0 |
| 0x1 - DITH4 | 23:4 |
| 0x2 - DITH5 | 23:5 |
| 0x3 - DITH6 | 23:6 (depicted) |

**Bits 5:0 – DITHERBUF[5:0]** Dithering Buffer Cycle Number
These bits represent the PER.DITHER bits buffer. When the double buffering is enabled, the value of this bit field is copied to the PER.DITHER bits on an UPDATE condition.
**Note:** This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [n:0] |
|---|---|
| 0x0 - NONE | - |
| 0x1 - DITH4 | 3:0 |
| 0x2 - DITH5 | 4:0 |
| 0x3 - DITH6 | 5:0 (depicted) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 49.8.21. Channel x Compare/Capture Buffer Value

**Name:** CCBUF
**Offset:** 0x70 + n*0x04 [n=0..5]
**Reset:** 0x00000000
**Property:** Write-Synchronized

CCBUFx is copied into CCx at TCC update time

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[17:10] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[9:2] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[1:0] | | DITHERBUF[5:0] | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:6 – CCBUF[17:0]** Channel x Compare/Capture Buffer Value
These bits hold the value of the Channel x Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCx). Accessing this register using the CPU or DMA will affect the corresponding CCBUFVx status bit.
**Notes:**

1. When the TCC is configured as a 16-bit timer/counter, the excess bits are read as zero.
2. This bit field occupies the MSB of the register, [23:m]. m is dependent on the Resolution bit in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [23:m] |
|---|---|
| 0x0 - NONE | 23:0 |
| 0x1 - DITH4 | 23:4 |
| 0x2 - DITH5 | 23:5 |
| 0x3 - DITH6 | 23:6 (depicted) |

**Bits 5:0 – DITHERBUF[5:0]** Dithering Buffer Cycle Number
These bits represent the CCx.DITHER bits buffer. When the double buffering is enable, DITHERBUF bits value is copied to the CCx.DITHER bits on an UPDATE condition.
**Note:** This bit field consists of the n LSB of the register. n is dependent on the value of the Resolution bits in the Control A register (CTRLA.RESOLUTION):

| CTRLA.RESOLUTION | Bits [n:0] |
|---|---|
| 0x0 - NONE | - |
| 0x1 - DITH4 | 3:0 |
| 0x2 - DITH5 | 4:0 |
| 0x3 - DITH6 | 5:0 (depicted) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 50. PTC - Peripheral Touch Controller

## 50.1. Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect a touch on the capacitive sensors. The external capacitive touch sensor is typically formed on a PCB, and the sensor electrodes are connected to the analog front end of the PTC through the I/O pins in the device. The PTC supports both self and mutual capacitance sensors.

In the Mutual Capacitance mode, sensing is done using capacitive touch matrices in various X-Y configurations, including indium tin oxide (ITO) sensor grids. The PTC requires one pin per X-line and one pin per Y-line.

In the Self Capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

The number of available pins and the assignment of X- and Y-lines is depending on both package type and device configuration. Refer to the Configuration Summary and I/O Multiplexing table for details.

**Related Links**

I/O Multiplexing and Considerations

## 50.2. Features

- Low-Power, High-Sensitivity, Environmentally Robust Capacitive Touch Buttons, Sliders, and Wheels
- Supports Wake-up on Touch from sleep mode Sleep mode
- Supports Mutual Capacitance and Self Capacitance Sensing
  - Mix-and-Match Mutual and Self Capacitance Sensors
- One Pin per Electrode – No External Components
- Load Compensating Charge Sensing
  - Parasitic capacitance compensation and adjustable gain for superior sensitivity
- Zero Drift Over the Temperature and VDDANA Range
  - Auto calibration and recalibration of sensors
- Single-shot and free-running Charge Measurement
- Hardware Noise Filtering and Noise Signal Desynchronization for High Conducted Immunity
- Selectable channel change delay allows choosing the settling time on a new channel, as required
- Acquisition-start triggered by command or through auto-triggering feature
- Low CPU utilization through interrupt on acquisition-complete
- Using ADC peripheral for signal conversion and acquisition

**Related Links**

I/O Multiplexing and Considerations


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 50.3. Block Diagram

**Figure 50-1.** PTC Block Diagram Mutual Capacitance

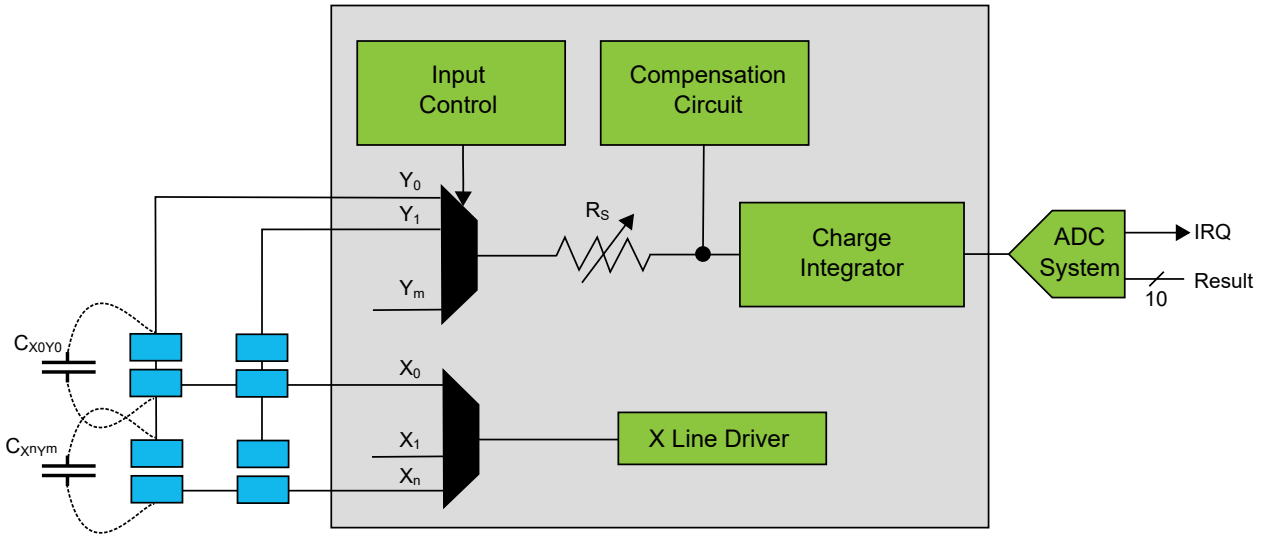
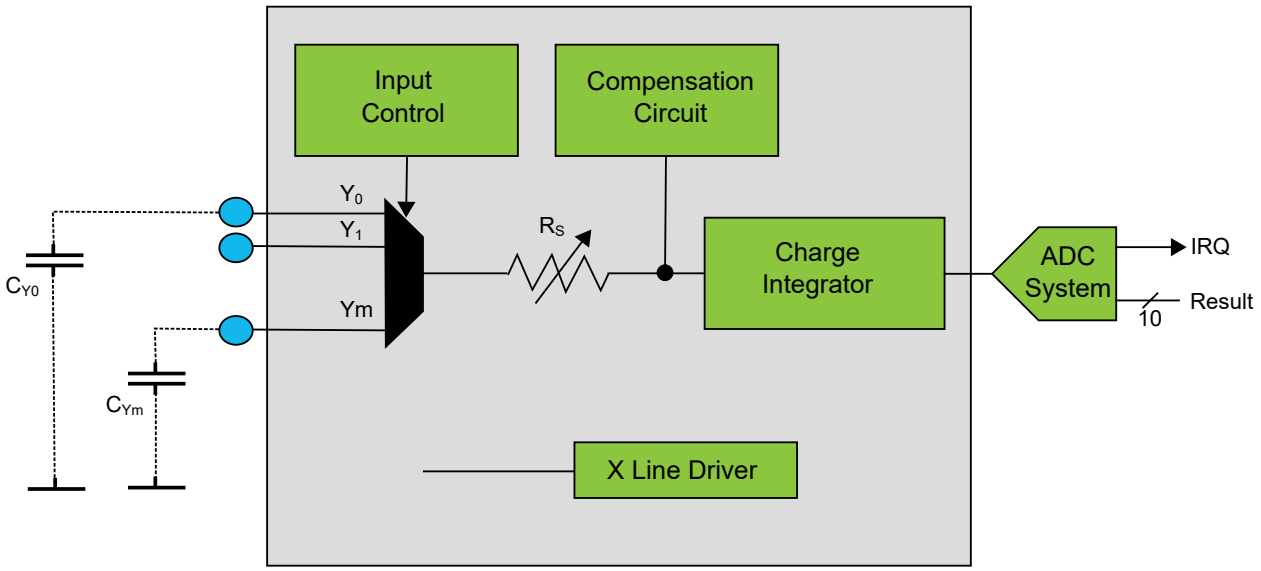



**Figure 50-2.** PTC Block Diagram Self Capacitance




## 50.4. Signal Description

**Table 50-1.** Signal Description for PTC

| Name | Type | Description |
|---|---|---|
| Y[m:0] | Analog | Y-line (Input/Output) |
| X[n:0] | Digital | X-line (Output) |

**Note:** The number of X- and Y-lines are device dependent. Refer to *Configuration Summary* for details.

Refer to *I/O Multiplexing and Considerations* for details on the pin mapping for this peripheral. One signal can be mapped on several pins.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
I/O Multiplexing and Considerations

## 50.5. System Dependencies

In order to use this peripheral, configure the other components of the system as described in the following sections.

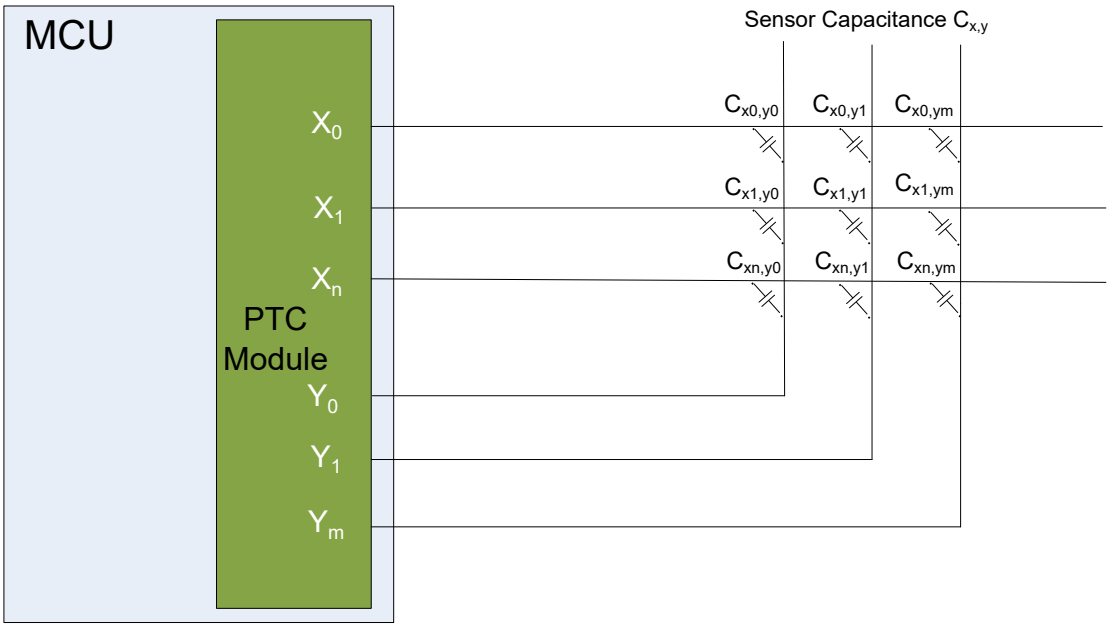
### 50.5.1. I/O Lines

The I/O lines used for analog X- and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1 kΩ or more can be used on X- and Y-lines.

#### 50.5.1.1. Mutual Capacitance Sensor Arrangement

A mutual capacitance sensor is formed between two I/O lines - an X electrode for transmitting and Y electrode for sensing. The mutual capacitance between the X and Y electrode is measured by the peripheral touch controller.

**Figure 50-3.** Mutual Capacitance Sensor Arrangement

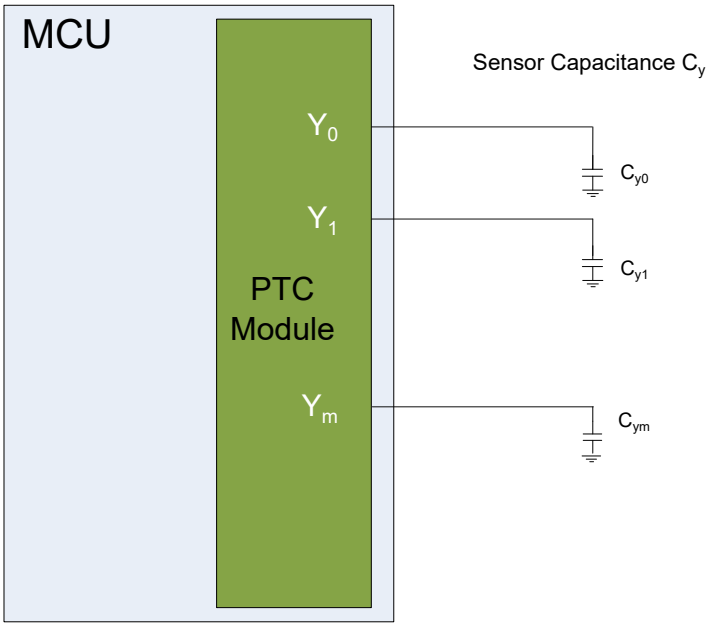



#### 50.5.1.2. Self Capacitance Sensor Arrangement

A self capacitance sensor is connected to a single pin on the peripheral touch controller through the Y electrode for sensing the signal. The sense electrode capacitance is measured by the peripheral touch controller.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 50-4.** Self-Capacitance Sensor Arrangement




For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide.

### 50.5.2. Analog-Digital Converter (ADC)

The PTC is using the ADC for signal conversion and acquisition. The ADC must be enabled and configured appropriately to allow correct behavior of the PTC.

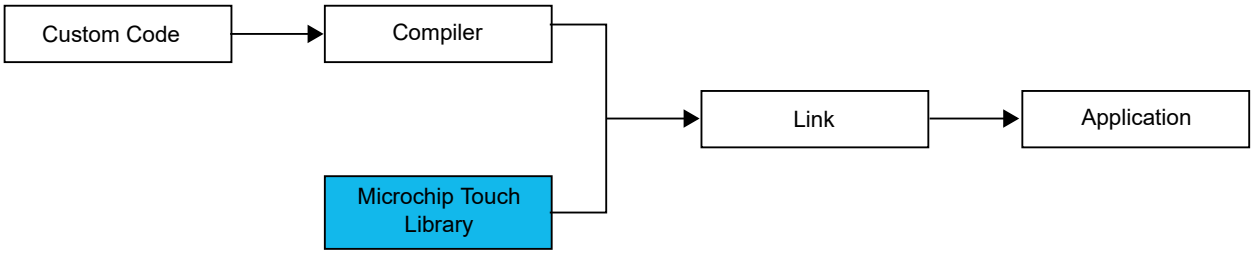
**Related Links**

ADC – Analog-to-Digital Converter

## 50.6. Functional Description

To access the PTC, the user must use MPLAB® Harmony v3 Touch Library. The MPLAB Harmony v3 Touch Library can be used to implement buttons, sliders, and wheels in a variety of combinations on a single interface.

**Figure 50-5.** Microchip Touch Library Usage




For additional information on Microchip Touch Library, refer to the “Touch Library Peripheral Touch Controller User Guide”, which is available for download at Microchip web site.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 51. Inter-IC ($I^2S$) Sound Controller

## 51.1. Overview

The Inter-IC Sound Controller ($I^2S$) provides bidirectional, synchronous and digital audio link with external audio devices.

This controller is compliant with the Inter-IC Sound ($I^2S$) bus specification. It supports TDM interface with external multi-slot audio codecs. It also supports Pulse Density Modulation (PDM) interface with external MEMS microphones.

The $I^2S$ consists of two Clock Units, one Transmit Serializer, and one Receive Serializer, that can be enabled separately, to provide Host, Client, or controller modes.

The pins associated with $I^2S$ peripheral are SDO,SDI, FSn, SCKn, and MCKn pins.

Peripheral DMAC channels, separate for each Serializer, allow a continuous high bitrate data transfer without processor intervention to the following:

- Audio codecs in Host, Client, or Controller mode
- Stereo DAC or ADC through dedicated $I^2S$ serial interface
- Multi-slot or multiple stereo DACs or ADCs, using the TDM format
- Mono or stereo MEMS microphones, using the PDM interface

Each Serializer supports using either a single DMAC channel for all data channels, or two separate DMAC channels for different data channels.

The $I^2S$ supports 8-bit and 16-bit compact stereo format. This helps in reducing the required DMA bandwidth by transferring the left and right samples within the same data word.

Usually, an external audio codec or digital signal processor (DSP) requires a clock which is a multiple of the sampling frequency *fs* (for example, 384×*fs*). The $I^2S$ peripheral in Host Mode and Controller mode is capable of outputting an output clock ranging from 16×*fs* to 1024×*fs* on the Host Clock pin (MCKn).

## 51.2. Features

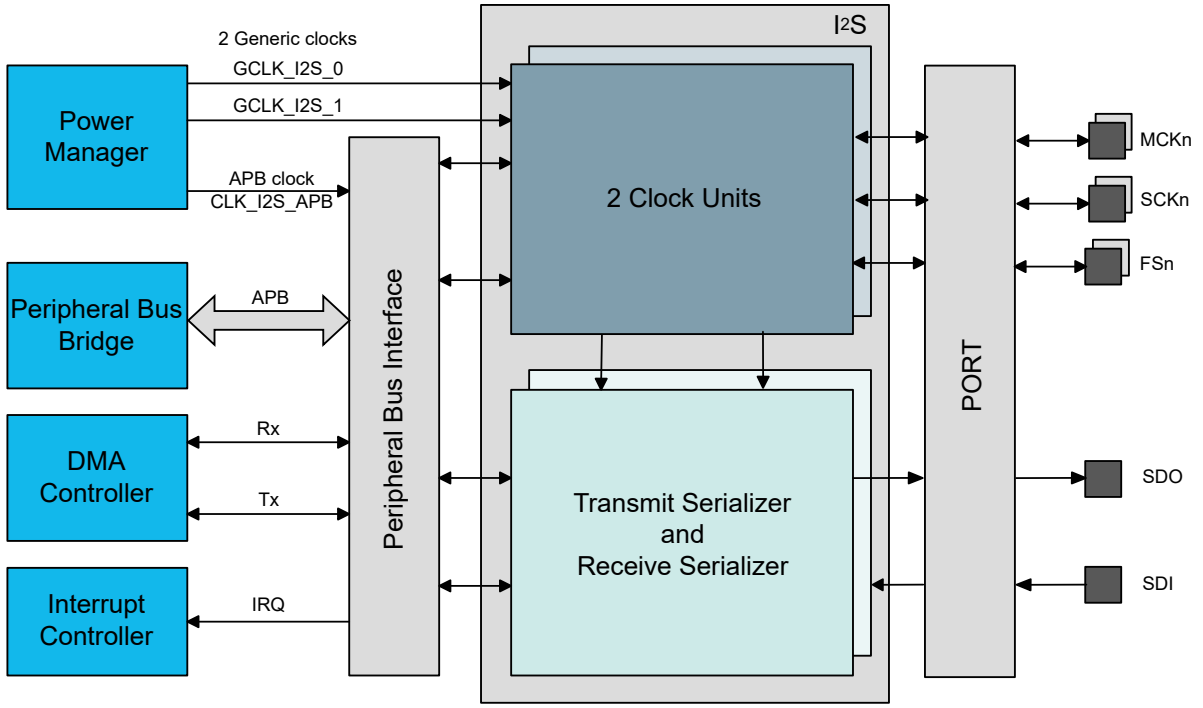
- Compliant with Inter-IC Sound ($I^2S$) bus specification
- Supported data formats:
  - 32-, 24-, 20-, 18-, 16-, and 8-bit mono or stereo format
  - 16- and 8-bit compact stereo format, with left and right samples packed in the same word to reduce data transfers
- Supported data frame formats:
  - 2-channel $I^2S$ with Word Select
  - 1- to 8-slot Time Division Multiplexed (TDM) with Frame Sync and individually enabled slots
  - 1- or 2-channel Pulse Density Modulation (PDM) reception for MEMS microphones
  - 1-channel burst transfer with non-periodic Frame Sync
- 2 independent Clock Units handling either the same clock or separate clocks for the Serializers:
  - Suitable for a wide range of sample frequencies *fs*, including 32kHz, 44.1kHz, 48kHz, 88.2kHz, 96kHz, and 192kHz
  - 16×*fs* to 1024×*fs* Host Clock generated for external audio CODECs
- Host, client, and controller modes:
  - Host: Data received/transmitted based on internally-generated clocks. Output Serial Clock on SCKn pin, Host Clock on MCKn pin, and Frame Sync Clock on FSn pin


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - Client: Data received/transmitted based on external clocks on Serial Clock pin (SCKn) or Host Clock pin (MCKn)
  - Controller: Only output internally generated Host clock (MCKn), Serial Clock (SCKn), and Frame Sync Clock (FSn)
- Individual enabling and disabling of Clock Units and Serializers
- DMA interfaces for each Serializer receiver or transmitter to reduce processor overhead:
  - Either one DMA channel for all data slots or
  - One DMA channel per data channel in stereo
- Smart Data Holding register management to avoid data slots mix after overrun or underrun

## 51.3. Block Diagram

**Figure 51-1.** I²S Block Diagram




## 51.4. Signal Description

**Table 51-1.**

| Pin Name | Pin Description | Type |
|---|---|---|
| MCKn | Host Clock for Clock Unit n | Input/Output |
| SCKn | Serial Clock for Clock Unit n | Input/Output |
| FSn | I²S Word Select or TDM Frame Sync for Clock Unit n | Input/Output |
| SDO | Serial Data Output for Transmit Serializer | Output |
| SDI | Serial Data Input for Receive Serializer | Input |

**Note:** One signal can be mapped on several pins.

**Related Links**

I/O Multiplexing and Considerations


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

## 51.5. Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

### 51.5.1. I/O Lines

Using the I²S I/O lines requires the I/O pins to be configured.

The I²S pins may be multiplexed with I/O Controller lines. The user must first program the I/O Controller to assign the desired I²S pins to their peripheral function. If the I²S I/O lines are not used by the application, they can be used for other purposes by the I/O Controller. It is required to enable only the I²S inputs and outputs actually in use.

**Related Links**

PORT - I/O Pin Controller

### 51.5.2. Power Management

The I²S will continue to operate in any sleep mode where the selected source clocks are running.

### 51.5.3. Clocks

The clock for the I²S bus interface (CLK_I2S_APB) is generated by the Power Manager. This clock is disabled at reset, and can be enabled in the Power Manager. It is recommended to disable the I²S before disabling the clock, to avoid freezing the I²S in an undefined state.

There are two generic clocks, GCLK_I2S_0 and GCLK_I2S_1, connected to the I²S peripheral, one for each I²S clock unit. The generic clocks (GCLK_I2S_n, n=0..1) can be set to a wide range of frequencies and clock sources. The GCLK_I2S_n must be enabled and configured before use.

The GCLK_I2S_n clocks must be enabled and configured before triggering Software Reset, so that the logic in all clock domains can be reset.

The generic clocks are only used in Host mode and Controller mode. In Host mode, the clock from clock unit 0 can be used for both Serializers to handle synchronous transfers, or a separate clock from different clock units can be used for each Serializer to handle transfers on non-related clocks.

**Related Links**

GCLK - Generic Clock Controller

### 51.5.4. DMA

The DMA request lines are connected to the DMA Controller (DMAC). Using the I²S DMA requests requires the DMA Controller to be configured first.

**Related Links**

DMAC – Direct Memory Access Controller

### 51.5.5. Interrupts

The interrupt request line is connected to the interrupt controller. Using I²S interrupts requires the interrupt controller to be configured first.

**Related Links**

Nested Vector Interrupt Controller

### 51.5.6. Events

Not applicable.

### 51.5.7. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodical service by the CPU through interrupts or similar,


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

improper operation or data loss may result during debugging. This peripheral can be forced to halt operation during debugging.

### 51.5.8. Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- DATAm
- INTFLAG
- SYNCBUSY

**Note:** Optional write protection is indicated by the "PAC Write Protection" property in the register description.

Write protection does not apply for accesses through an external debugger.

### 51.5.9. Analog Connections

Not applicable.

## 51.6. Functional Description

### 51.6.1. Principle of Operation

The $I^2S$ uses three or four communication lines for synchronous data transfer:

- SDO output for Transmit Serializer
- SDI input for Receive Serializer
- SCKn for the serial clock in Clock Unit n (n = 0..1)
- FSn for the frame synchronization or $I^2S$ word select, identifying the beginning of each frame
- Optionally, MCKn to output an oversampling clock to an external codec

$I^2S$ data transfer is frame based, where a serial frame has the following characteristics:

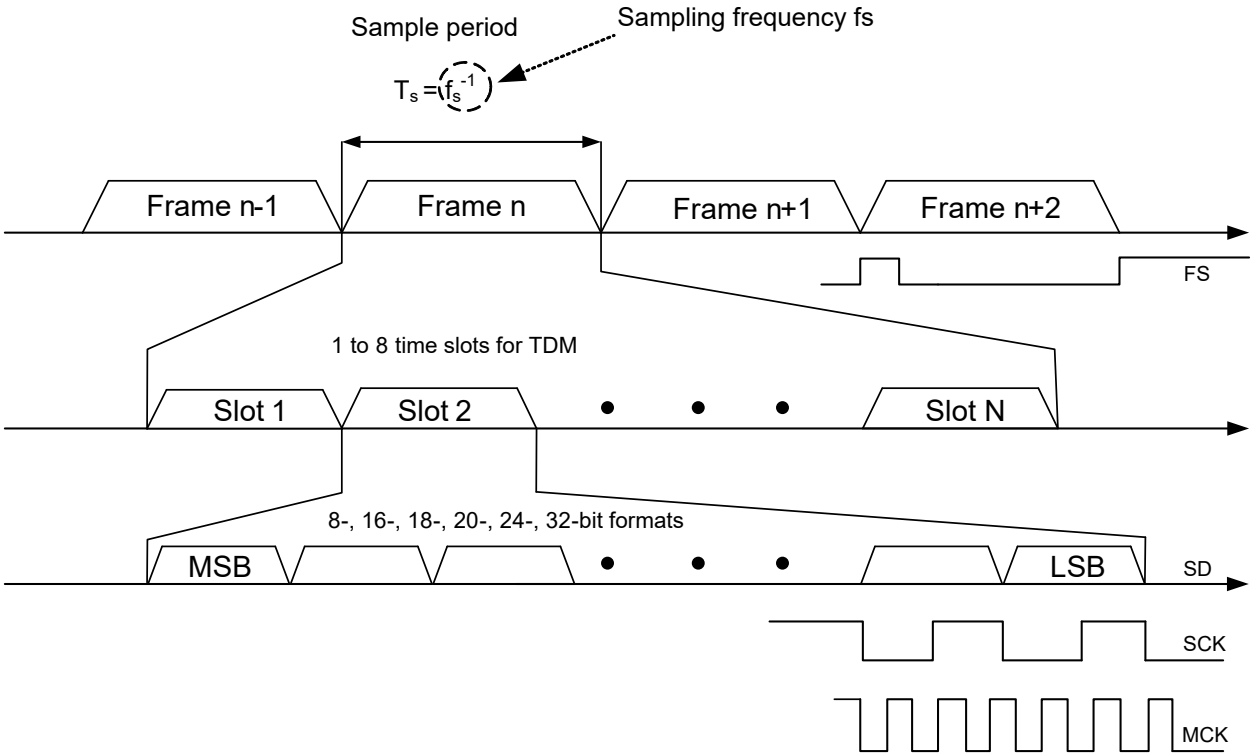
- Starts with the frame synchronization active edge
- Consists of 1 to 8 data slots, that are 8, 16, 24, or 32 bits wide.

Each data slot is used to transfer one data sample of 8, 16, 18, 20, 24 or 32 bits.

Frame based data transfer is described in the following figure:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 51-2.** Data Format: Frames, Slot, Bits and Clocks




I²S supports multiple data formats such as:

- 32, 24, 20, 18, 16, and 8 bits mono or stereo format
- 16-bit and 8-bit compact stereo format, with left and right samples packed in the same word to reduce data transfers

In mono format, Transmit mode, data written to the left channel is duplicated to the right output channel. In mono format, Receiver mode, data received from the right channel is ignored and data received from the left channel is duplicated in to the right channel.

In mono format, TDM Transmit mode with more than two slots, data written to the even-numbered slots is duplicated in to the following odd-numbered slot.

In mono format, TDM Receiver mode with more than two slots, data received from the even-numbered slots is duplicated in to the following odd-numbered slot.

Mono format can be enabled by writing a '1' to the MONO bit in the Serializer m Control register (SERCTRLm.MONO).

I²S support different data frame formats:

- 2-channel I²S with Word Select
- 1-slot to 8-slot Time Division Multiplexed (TDM) with Frame Sync and individually enabled slots
- 1-channel or 2-channel Pulse Density Modulation (PDM) reception for MEMS microphones
- 1-channel burst transfer with non-periodic Frame Sync

In 2 channel I²S mode, number of slots configured is one or two and successive data words corresponds to left and right channel. Left and right channel are identified by polarity of Word Select signal (FSn signal). Each frame consists of one or two data word(s). In the case of compact stereo format, the number of slots can be one. When 32-bit slot size is used, the number of slots can be two.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

In TDM format, number slots can be configured up to 8 slots. If 4 slots are configured, each frame consists of 4 data words.

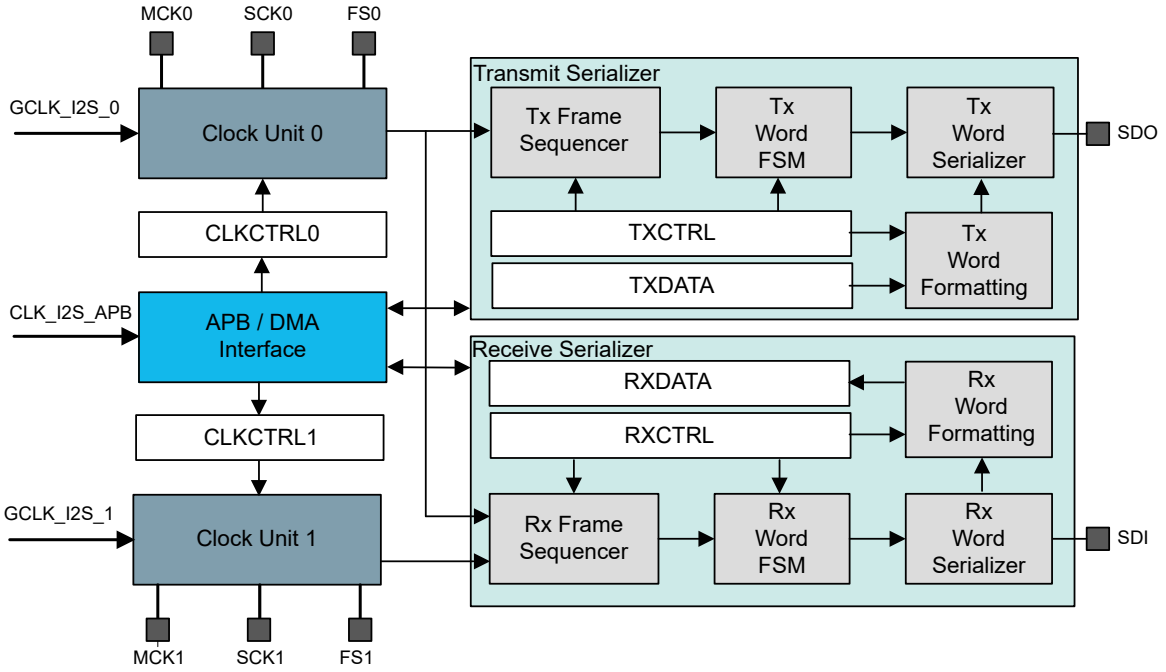
In PDM format, continuous 1-bit data samples are available on the SDI line for each SCKn rising and SCKn falling edge as in case of a MEMS microphone with PDM interface.

1-channel burst transfer with non-periodic Frame Sync mode is useful typically for passing control non-auto data as in case of DSP. In Burst mode, a single Data transfer starts at each Frame Sync pulse, and these pulses are 1-bit wide and occur only when a Data transfer is requested.

**Note:** Compact stereo modes (16C and 8C) are not supported in the Burst mode.

Sections I2S Format - Reception and Transmission Sequence with Word Select, TDM Format - Reception and Transmission Sequence and I2S Application Examples describe more about frame/ data formats and register settings required for different I²S applications.

**Figure 51-3.** I²S Functional Block Diagram




### 51.6.1.1. Initialization

The I²S features two Clock Units, one Transmit Serializer, and One Receive Serializer. The Transmit Serializer uses Clock Unit 0, while the Receive Serializer can either share the same Clock Unit 0 or use the Clock Unit 1.

Before enabling the I²S, the following registers must be configured:

- Clock Control registers (CLKCTRLn)
- Serializer Control registers (TXCTRL and/or RXCTRL)

In Host mode, one of the generic clocks for the I²S must also be configured to operate at the required frequency, as described in Principle of Operation.

- *fs* is the sampling frequency that defines the frame period
- CLKCTRLn.NBSLOTS defines the number of slots in each frame
- CLKCTRLn.SLOTSIZE defines the number of bits in each slot
- SCKn frequency must be $f_{SCKn} = fs \times$ number_of_slots × number_of_bits_per_slot)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Once the configuration has been written, the I²S Clock Units and Serializers can be enabled by writing a '1' to the CKENn, TXEN, and/or RXEN bits and to the ENABLE bit in the Control register (CTRLA). The Clock Unit n can be enabled alone, in Controller Mode, to output clocks to the MCKn, SCKn, and FSn pins. The Clock Units must be enabled if Serializers are enabled.

The Clock Units, the Transmit Serializer and the Receive Serializer can be disabled independently by writing a '0' to CTRLA.CKENn, CTRLA.TXEN, and CTRLA.RXEN, respectively. Once requested to stop, they will only stop when the pending transmit frames will be completed, if any. When requested to stop, the ongoing reception of the current slot will be completed and then the Serializer will be stopped.

**Example 51-1.** *Example Requirements: fs*=48kHz, MCKn=384×*fs*

If a 384×*fs* MCKn Host Clock is required (i.e. 18.432 MHz), the I²S generic clock could run at 18.432 MHz with a Host Clock Output Division Factor of 1 (selected by writing CLKCTRLn.MCKOUTDIV=0x0) in order to obtain the desired MCKn frequency.

When using 6 slots per frame (CLKCTRLn.NBSLOTS=0x5) and 32-bit slots (CLKCTRLn.SLOTSIZE=0x3), the desired SCKn frequency is

$f_{SCKn}$ = 48 kHz × 6 × 32 = 9.216 MHz

This frequency can be achieved by dividing the I²S generic clock output of 18.432 MHz by factor 2: Writing CLKCTRLn.MCKDIV=0x1 will select the correct division factor and output the desired SCKn frequency of 9.216 MHz to the SCKn pin.

If MCKn is not required, the generic clock could be set to 9.216 MHz and CLKCTRLn.MCKDIV=0x0.

### 51.6.2. Basic Operation

The Receiver can be operated by reading the Receive Data Holding register (RXDATA), whenever the Receive Ready m bit in the Interrupt Flag Status and Clear register (INTFLAG.RXRDYm) is set. Successive values read from the RXDATA register will correspond to the samples from the left and right audio channels. In TDM mode, the successive values read from RXDATA correspond to the first slot to the last slot. For instance, if I²S is configured in TDM mode with 4 slots in a frame, then successive values written to RXDATA register correspond to first, second, third, and fourth slot. The number of slots in TDM is configured in CLKCTRLn.NBSLOTS.

The Transmitter can be operated by writing to the Transmit Data Holding register (TXDATA), whenever the Transmit Ready m bit in the Interrupt Flag Status and Clear register (INTFLAG.TXRDYm) is set. Successive values written to TXDATA register should correspond to the samples from the left and right audio channels. In TDM mode, the successive values written to TXDATA correspond to the first, second, third, slot to the last slot. The number of slots in TDM is configured in CLKCTRLn.NBSLOTS.

The Receive Ready and Transmit Ready bits can be polled by reading the INTFLAG register.

The processor load can be reduced by enabling interrupt-driven operation. The RXRDYm and/or TXRDYm interrupt requests can be enabled by writing a '1' to the corresponding bit in the Interrupt Enable register (INTENSET). The interrupt service routine associated to the I²S interrupt request will then be executed whenever Receive Ready or Transmit Ready status bits are set.

The processor load can be reduced further by enabling DMA-driven operation. Then, the DMA channels support up to four trigger sources from the I²S peripheral. These four trigger sources in DMAC channel are

- I2S RX 0,
- I2S RX 1,


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

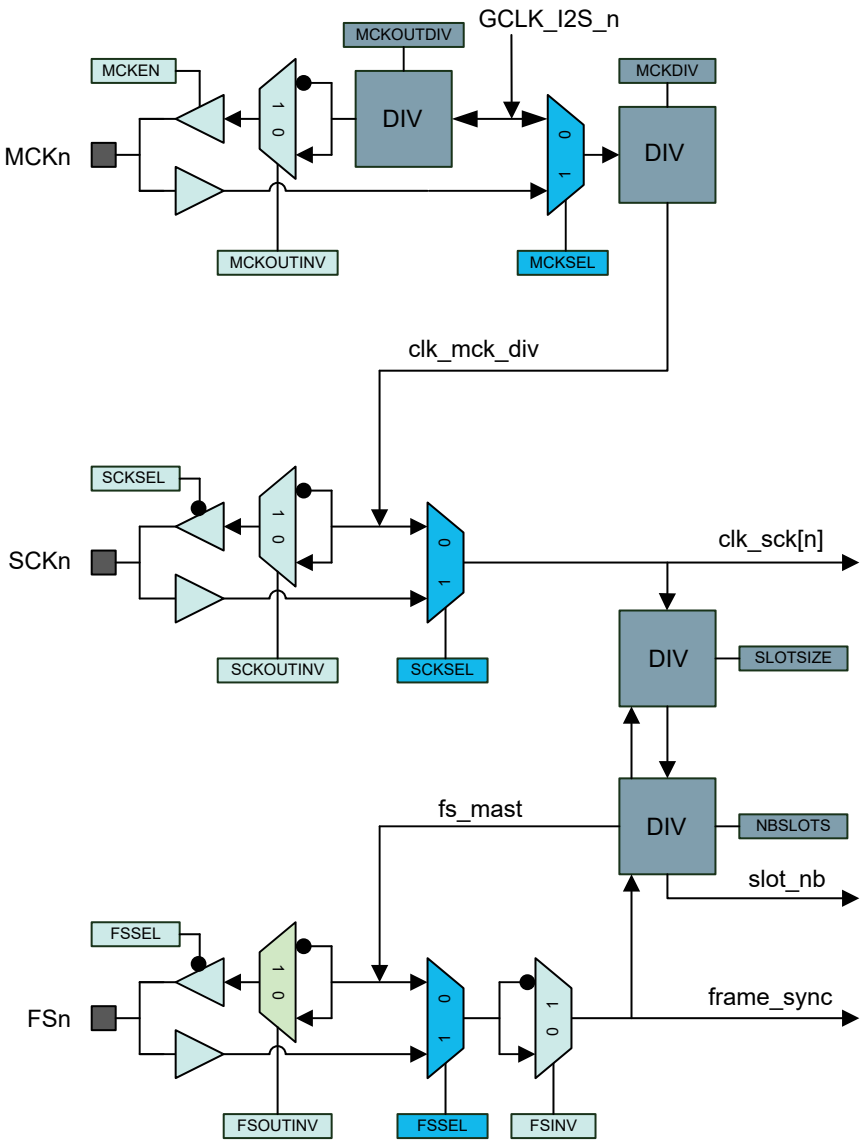
- I2S TX 0, and
- I2S TX 1.

For further reference, these are called I2S_DMAC_ID_RX_m and I2S_DMAC_ID_TX_m triggers (m=0..1). By using these trigger sources, one DMA data transfer will be executed whenever the Receive Ready or Transmit Ready status bits are set.

### 51.6.2.1. Host Clock, Serial Clock, and Frame Sync Generation

The generation of clocks in the I²S is described in the next figure.

**Figure 51-4.** I²S Clocks




Generation

#### 51.6.2.1.1. Client Mode

In Client mode, the Serial Clock and Frame Sync (Word Select in I²S mode and Frame Sync in TDM mode) are driven by an external host. SCKn and FSn pins are inputs and no generic clock is required by the I²S.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 51.6.2.1.2. Host Mode and Controller Mode

In Host Mode, the Host Clock (MCKn), the Serial Clock (SCKn), and the Frame Sync Clock (FSn) are generated by the I²S controller. The user can configure the Host Clock, Serial Clock, and Word Select Frame Sync signal (Word Select in I²S mode and Frame Sync in TDM mode) using the Clock Unit n Control register (CLKCTRLn). MCKn, SCKn, and FSn pins are outputs and a generic clock is used to derive the I²S clocks.

In some applications, audio CODECs connected to the I²S pins may require a Host Clock signal with a frequency multiple of the audio sample frequency *fs*, such as 256×*fs*.

In Controller mode, only the Clock generation unit needs to be configured by writing to the CTRLA and CLKCTRLn registers, where parameters such as clock division factors, Number of slots, Slot size, Frame Sync signal, clock enable are selected.

#### 51.6.2.1.3. MCKn Clock Frequency

When the I²S is in Host mode, writing a '1' to CLKCTRLn.MCKEN will output GCLK_I2S_n as Host Clock to the MCKn pin. The Host Clock to MCKn pin can be divided by writing to CLKCTRLn.MCKSEL and CLKCTRLn.MCKOUTDIV. The Host Clock (MCKn) frequency is GCLK_I2S_n frequency divided by (MCLKOUTDIV+1).

$$f(\mathrm{MCKn}) = \frac{f(\mathrm{GCLK_I2S_}n)}{(\mathrm{MCKOUTDIV}+1)}$$

#### 51.6.2.1.4. SCKn Clock Frequency

When the Serial Clock (SCKn) is generated from GCLK_I2S_n and both CLKCTRLn.MCKSEL and CLKCTRLn.SCKSEL are zero, the Serial Clock (SCKn) frequency is GCLK_I2S_n frequency divided by (MCKDIV+1).

i.e.

$$f(\mathrm{SCKn}) = \frac{f(\mathrm{GCLK_I2S_}n)}{(\mathrm{MCKDIV}+1)}$$

#### 51.6.2.1.5. Relation Between MCKn, SCKn, and Sampling Frequency *fs*

Based on sampling frequency *fs*, the SCKn frequency requirement can be calculated:

- SCKn frequency: $f_{\mathrm{SCKn}} = fs \times \mathrm{total_number_of_bits_per_frame}$,
- Where $\mathrm{total_number_of_bits_per_frame} = \mathrm{number_of_slots} \times \mathrm{number_of_bits_per_slots}$.
- The number of slots is selected by writing to the Number of Slots in Frame bit field in the Clock Unit n Control (CLKCTRLn) register: $\mathrm{number_of_slots} = \mathrm{NBSLOTS} + 1$.
- The number of bits per slot (8, 16, 24, or 32 bit) is selected by writing to the Slot Size bit field in CLKCTRLn: .
- Consequently, $f_{\mathrm{SCKn}} = 8 \times fs \times (\mathrm{NBSLOTS}+1) \times (\mathrm{SLOTSIZE}+1)$.

The clock frequencies $f_{\mathrm{SCKn}}$ and $f_{\mathrm{MCKn}}$ are derived from the generic clock frequency $f_{\mathrm{GCLK_I2S_n}}$ :

- $f_{\mathrm{GCLK_I2S_n}} = f_{\mathrm{SCKn}} \times (\mathrm{CLKCTRLn.MCKDIV}+1)$
  $= 8 \times fs \times (\mathrm{NBSLOTS}+1) \times (\mathrm{SLOTSIZE}+1) \times (\mathrm{MCKDIV}+1)$
  , and
- $f_{\mathrm{GCLK_I2S_n}} = f_{\mathrm{MCKn}} \times (\mathrm{MCKOUTDIV}+1)$.

Substituting the right hand sides of the two last equations yields:

$$f_{\mathrm{MCKn}} = \frac{f_{\mathrm{GCLK_I2S_n}}}{\mathrm{MCKOUTDIV}+1}$$

$$f_{\mathrm{MCKn}} = \frac{8 \cdot fs(\mathrm{SLOTSIZE}+1) \cdot (\mathrm{NBSLOTS}+1) \cdot (\mathrm{MCKDIV}+1)}{\mathrm{MCKOUTDIV}+1}$$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

If a Host Clock output is not required, the GCLK_I2S generic clock can be configured as SCKn by writing a '0'to CLKCTRLn.MCKDIV. Alternatively, if the frequency of the generic clock is a multiple of the required SCKn frequency, the MCKn-to-SCKn divider can be used with the ratio defined by writing the CLKCTRLn.MCKDIV field.

The FSn pin is used as Word Select in I$^2$S format and as Frame Synchronization in TDM format, as described in I2S Format - Reception and Transmission Sequence with Word Select and TDM Format - Reception and Transmission Sequence, respectively.

#### 51.6.2.2. Data Holding Registers

For both the Transmit and the Receive Serializer, the I$^2$S user interface includes a Data register (TXDATA and RXDATA, respectively). They are used to access data samples for all data slots.

##### 51.6.2.2.1. Data Reception Mode

In receiver mode, the RXDATA register stores the received data.

When a new data word is available in the RXDATA register, the Receive Ready bit (RXRDYm) in the Interrupt Flag Status and Clear register (INTFLAG) is set. Reading the RXDATA register will clear this bit.

A receive overrun condition occurs if a new data word becomes available before the previous data word has been read from the RXDATA register. Then, the Receive Overrun bit in INTFLAG will be set (INTFLAG.RXORm). This interrupt can be cleared by writing a '1' to it.

##### 51.6.2.2.2. Data Transmission Mode

In Transmitter mode, the TXDATA register contains the data to be transmitted.

when TXDATA is empty, the Transmit Ready bit in the Interrupt Flag Status and Clear register is set (INTFLAG.TXRDYm). Writing to TXDATA will clear this bit.

A transmit underrun condition occurs if data present in TXDATA is sent and no new data is written to TXDATA register before the next time slot. Then, the Transmit Underrun bit in INTFLAG will be set (INTFLAG.TXURm). This interrupt can be cleared by writing a '1' to it. The Transmit Data when Underrun bit in the Tx Serializer Control register (TXCTRL.TXSAME) configures whether a zero data word is transmitted in case of underrun (TXCTRL.TXSAME=0), or the previous data word for the current transmit slot number is transmitted again (TXCTRL.TXSAME=1).

### 51.6.3. Host, Controller, and Client Modes

In Host and Controller modes, the I$^2$S provides the Serial Clock, a Word Select/Frame Sync signal and optionally a Host Clock.

In Controller mode, the I$^2$S Serializers are disabled. Only the clocks are enabled and output for external receivers and/or transmitters.

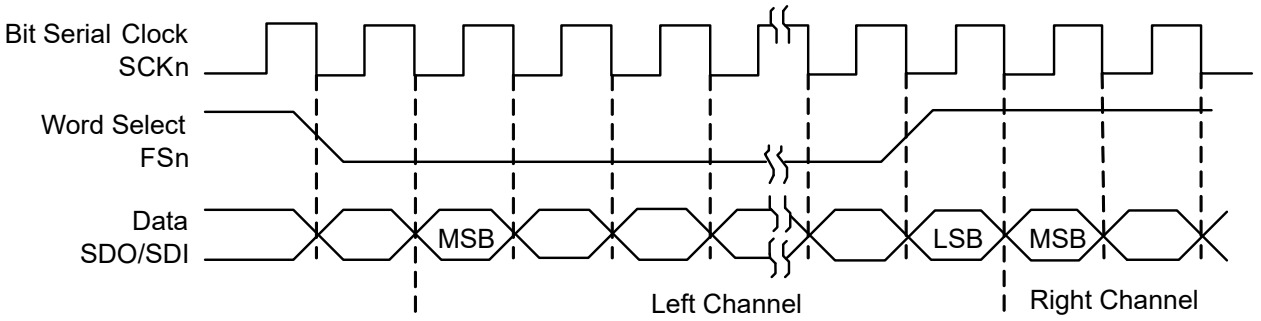
In Client mode, the I$^2$S receives the Serial Clock and the Word Select/Frame Sync Signal from an external host. SCKn and FSn pins are inputs.

### 51.6.4. I$^2$S Format - Reception and Transmission Sequence with Word Select

As specified in the I$^2$S protocol, data bits are left-adjusted in the Word Select slot, with the MSB transmitted first, starting one clock period after the transition on the Word Select line.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 51-5.** I²S Reception and Transmission Sequence




Data bits are sent on the falling edge of the Serial Clock and sampled on the rising edge of the Serial Clock. The Word Select line indicates the channel in transmission, a low level for the left channel and a high level for the right channel.

In I²S format, typical configurations are described below. These configurations do not list all necessary settings, but only basic ones. Other configuration settings are to be done as per requirement such as clock and DMA configurations.

**Case 1: I²S 16-bit compact stereo receiver**

- Slot size configured as 16 bits (CLKCTRL0.SLOTSIZE = 0x1)
- Number of slots configured as 2 (CLKCTRL0.NBSLOTS = 0x1)
- Data size configured as 16-bit compact stereo (RXCTRL.DATASIZE = 0x05)
- Data delay from Frame Sync configured as 1-bit delay (CLKCTRLn.BITDELAY = 0x01)
- Frame Sync Width configured as HALF frame (CLKCTRLn.FSWIDTH = 0x01)

**Case 2: I²S 24-bit stereo Transmitterwith 24-bit slot**

- Slot size configured as 24 bits (CLKCTRL0.SLOTSIZE = 0x2)
- Number of slots configured as 2 (CLKCTRL0.NBSLOTS = 0x1)
- Data size configured as 24 bits (TXCTRL.DATASIZE = 0x01)
- Data delay from Frame Sync configured as 1-bit delay (CLKCTRLn.BITDELAY = 0x01)
- Frame Sync Width configured as HALF frame (CLKCTRLn.FSWIDTH = 0x01)

In both cases, it will ensure that Word select signal is 'low level' for the left channel and 'high level' for the right channel.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing the Data Word Size bit group in the Serializer Control register (RXCTRL.DATASIZE or TXCTRL.DATASIZE, respectively).

If the slot allows for more data bits than the number of bits specified in the respective DATASIZE field, additional bits are appended to the transmitted or received data word as specified in the RXCTRL/TXCTRL.EXTEND field. If the slot allows less data bits than programmed, the extra bits are not transmitted, or received data word is extended based on the EXTEND field value.

### 51.6.5. TDM Format - Reception and Transmission Sequence

In Time Division Multiplexed (TDM) format, the number of data slots sent or received within each frame will be (CLKCTRLn.NBSLOTS + 1).

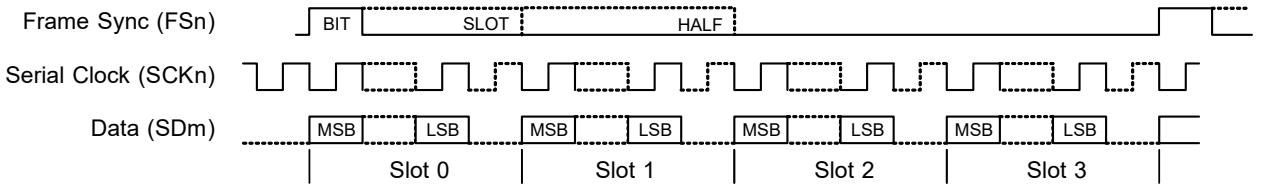
By configuring the CLKCTRLn register (CLKCTRLn.FSWIDTH and CLKCTRLn.FSINV), the Frame Sync pulse width and polarity can be modified.

By configuring RXCTRL and/or TXCTRL, data bits can be left-adjusted or right-adjusted in the slot. It can also configure the data transmission/reception with either the MSB or the LSB transmitted/


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

received first and starting the transmission/reception either at the transition of the FSn pin or one clock period after.

**Figure 51-6.** TDM Format Reception and Transmission Sequence




Data bits are sent on the falling edge of the Serial Clock and sampled on the rising edge of the Serial Clock. The FSn pin provides a frame synchronization signal, at the beginning of slot 0. The delay between the frame start and the first data bit is defined by writing the CLKCTRLn.BITDELAY field.

The Frame Sync pulse can be either one SCKn period (BIT), one slot (SLOT), or one half frame (HALF). This selection is done by writing the CLKCTRLn.FSWIDTH field.

The number of slots is selected by writing the CLKCTRLn.NBSLOTS field.

The number of bits in each slot is selected by writing the CLKCTRLn.SLOTSIZE field.

The length of transmitted words can be chosen among 8, 16, 18, 20, 24, and 32 bits by writing the DATASIZE field in the Serializer Control register (RXCTRL and/or TXCTRL).

If the slot allows more data bits than the number of bits specified in the RXCTRL. and/or TXCTRL.DATASIZE bit field, additional bits are appended to the transmitted or received data word as specified in the RXCTRL. and/or TXCTRL.EXTEND bit field. If the slot allows less data bits than programmed, the extra bits are not transmitted, or received data word is extended based on the EXTEND field value.

### 51.6.6. PDM Reception

In Pulse Density Modulation (PDM) reception mode, continuous 1-bit data samples are available on the SDI line on each SCKn rising edge, e.g. by a MEMS microphone with PDM interface. When using two channel PDM microphones, the second one (right channel) is configured to output data on each SCKn falling edge.

For one PDM microphone, the $I^2S$ controller should be configured in normal Receive mode with one slot and 16- or 32-bit data size, so that 16 or 32 samples of the microphone are stored into each data word.

For two PDM microphones, the $I^2S$ controller should be configured in PDM2 mode with one slot and 32-bit data size. The Rx Serializer will store 16 samples of each microphone in one half of the data word, with left microphone bits in lower half and right microphone bits in upper half, like in compact stereo format.

Based on oversampling frequency requirement from PDM microphone, the SCKn frequency must be configured in the $I^2S$ controller.

> A microphone that requires a sampling frequency of $fs$ = 48 kHz and an oversampling frequency of $fo$=64 × $fs$ would require an SCKn frequency of 3.072 MHz.

After selecting a proper frequency for GCLK_I2S_n and according Host Clock Division Factor in the Clock Unit n Control register (CLKCTRLn.MCKDIV), SCKn must be selected as per required frequency.

In PDM mode, only the clock and data line (SCKn and SDIn) pins are used.

To configure PDM2 mode, set SLOTSIZE = 0x01 (16-bits), NBSLOTS = 0x00 (1 slots) and RXCTRL.DATASIZE = 0x00 (32-bit).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.6.7. Data Formatting Unit

To allow more flexibility, data words received by the Receive Serializer will be formatted by the Receive Formatting Unit before being stored into the Data Holding register (DATAm). The data words written into DATAm register will be formatted by the Transmit Formatting Unit before transmission by the Transmit Serializer .

The formatting options are defined in RXCTRL and TXCTRL:

- SLOTADJ for left or right justification in the slot
- BITREV for bit reversal
- WORDADJ for left or right justification in the data word
- EXTEND for extension to the word size

### 51.6.8. DMA, Interrupts and Events

**Table 51-2.** Module Request for $I^2S$

| Condition | DMA request | DMA request is cleared | Interrupt request | Event input/ output |
|---|---|---|---|---|
| Receive Ready | YES | When data is read | YES | |
| Transmit Ready (Buffer empty) | YES | When data is written | YES | |
| Receive Overrun | | | YES | |
| Transmit Underrun | | | YES | |

#### 51.6.8.1. DMA Operation

Each Serializer can be connected either to one single DMAC channel or to one DMAC channel per data slot in Stereo mode. This is selected by writing the RXCTRL/TXCTRL.DMA bit.

**Table 51-3.** $I^2C$ DMA Request Generation

| SERCTRLm.DMA | Mode | Slot Parity | DMA Request Trigger |
|---|---|---|---|
| 0 | Receiver | all | I2S_DMAC_ID_RX_m |
| | Transmitter | all | I2S_DMAC_ID_TX_m |
| 1 | Receiver | even | I2S_DMAC_ID_RX_m |
| | | odd | I2S_DMAC_ID_TX_m |
| | Transmitter | even | I2S_DMAC_ID_TX_m |
| | | odd | I2S_DMAC_ID_RX_m |

The DMAC reads from the RXDATA register and writes to the TXDATA register for all data slots, successively.

The DMAC transfers may use 32-bit, 16-bit, or or 8-bit transactions according to the value of the TXCTRL/RXCTRL.DATASIZE field. 8-bit compact stereo uses 16-bit and 16-bit compact stereo uses 32-bit transactions.

#### 51.6.8.2. Interrupts

The $I^2S$ has the following interrupt sources:

- Receive Ready (RXRDYm): This is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.
- Receive Overrun (RXORm): This is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.
- Transmit Ready (TXRDYm): This is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.
- Transmit Underrun (TXURm): This is an asynchronous interrupt and can be used to wake-up the device from any sleep mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear (INTFLAG) register is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a one to the corresponding bit in the Interrupt Enable Set (INTENSET) register, and disabled by writing a one to the corresponding bit in the Interrupt Enable Clear (INTENCLR) register. An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt is disabled, or the $I^2S$ is reset. Refer to the INTFLAG register for details on how to clear interrupt flags. All interrupt requests from the peripheral are ORed together on system level to generate one combined interrupt request to the NVIC. Refer to the *"Nested Vector Interrupt Controller"* for details. The user must read the INTFLAG register to determine which interrupt condition is present.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**

Nested Vector Interrupt Controller

#### 51.6.8.3. Events

Not applicable.

### 51.6.9. Sleep Mode Operation

The $I^2S$ continues to operate in all sleep modes that still provide its clocks.

### 51.6.10. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

When executing an operation that requires synchronization, the corresponding Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY) will be set immediately, and cleared when synchronization is complete.

If an operation that requires synchronization is executed while the corresponding SYNCBUSY bit is '1', a peripheral bus error is generated.

The following bits are synchronized when written:

- Software Reset bit in the Control A register (CTRLA.SWRST). SYNCBUSY.SWRST is set to '1' while synchronization is in progress.
- Enable bit in the Control A register (CTRLA.ENABLE). SYNCBUSY.ENABLE is set to '1' while synchronization is in progress.
- Clock Unit x Enable bits in the Control A register (CTRLA.CKENx). SYNCBUSY.CKENx is set to '1' while synchronization is in progress.
- Serializer Enable bits in the Control A register (CTRLA.TXEN and CTRLA.RXEN). SYNCBUSY.TXEN/RXEN is set to '1' while synchronization is in progress.

The following registers require synchronization when read or written:

- Transmit Data register (TXDATA) is Write-Synchronized. SYNCBUSY.TXDATA is set to '1' while synchronization is in progress.
- Receive Data register (RXDATA) is Read-Synchronized. SYNCBUSY.RXDATA is set to '1' while synchronization is in progress.

Synchronization is denoted by the Read-Synchronized or Write-Synchronized property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.6.11. Loop-Back Mode

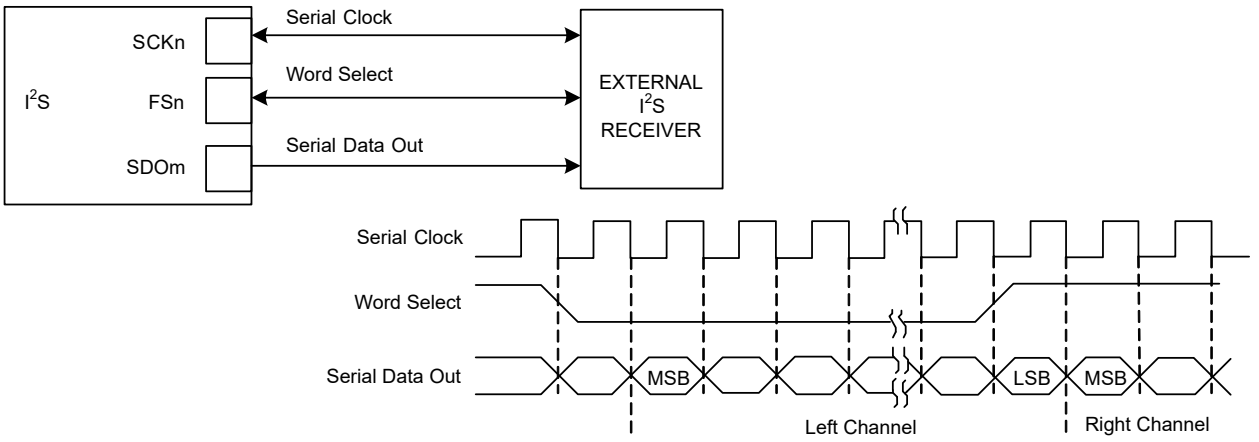
For debugging purposes, the I²S can be configured to loop back the Transmitter to the Receiver. Writing a '1' to the Loop-Back Test Mode bit in the Rx Serializer Control register (RXCTRL.RXLOOP)will connect SDO to SDI, so that transmitted data is also received.

Writing RXCTRL.RXLOOP=0 will restore the normal behavior and connection between Receive Serializer and SDI pin input. As for other changes to the Serializers configuration, the Receive Serializer must be disabled before writing the TXCTRL register to update TXCTRL.RXLOOP.

## 51.7. I²S Application Examples

The I²S can support several serial communication modes used in audio or high-speed serial links. Some standard applications are shown in the following figures.
**Note:** The following examples are not a complete list of serial link applications supported by the I²S.

**Figure 51-7.** Audio Application Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 51-8.** Time Slot Application Block Diagram

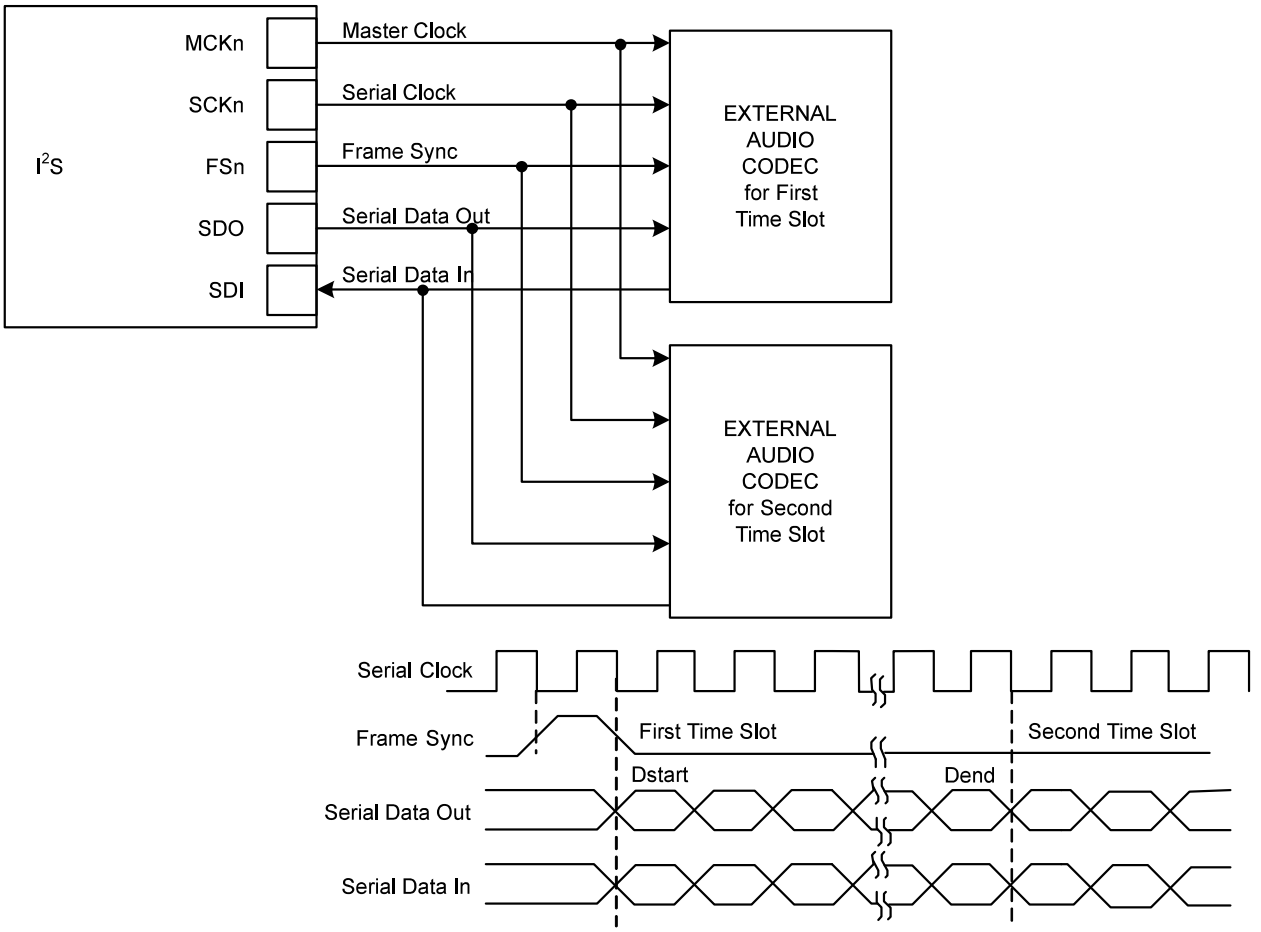
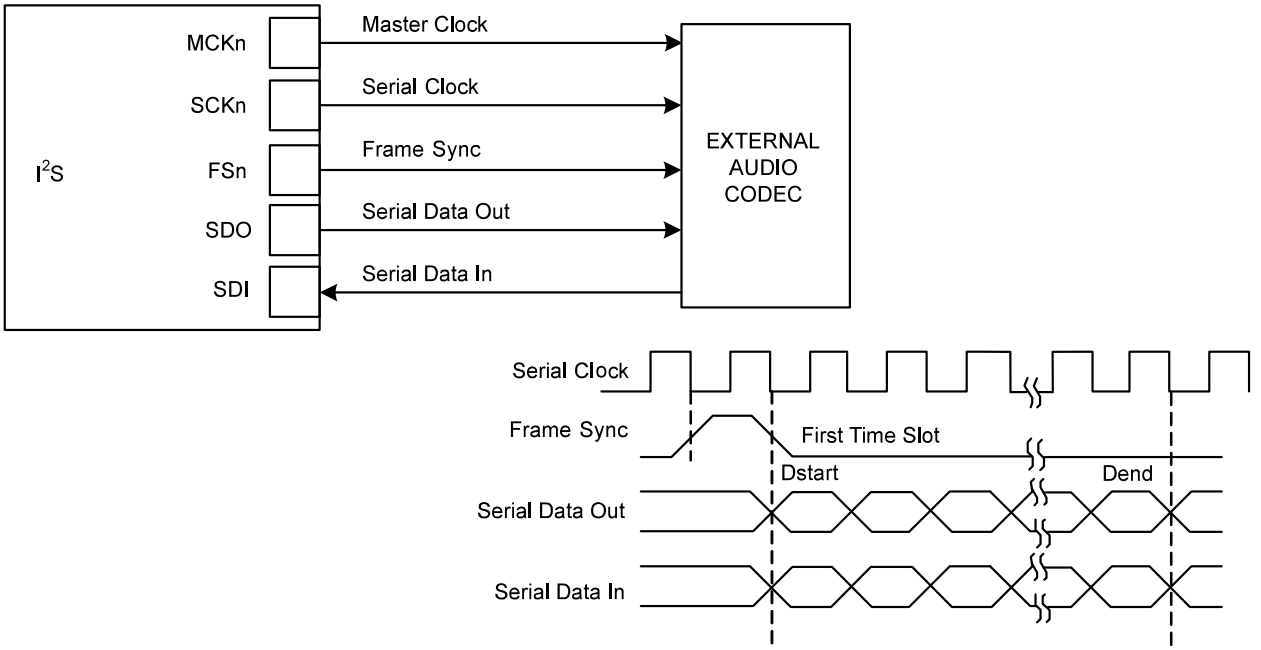


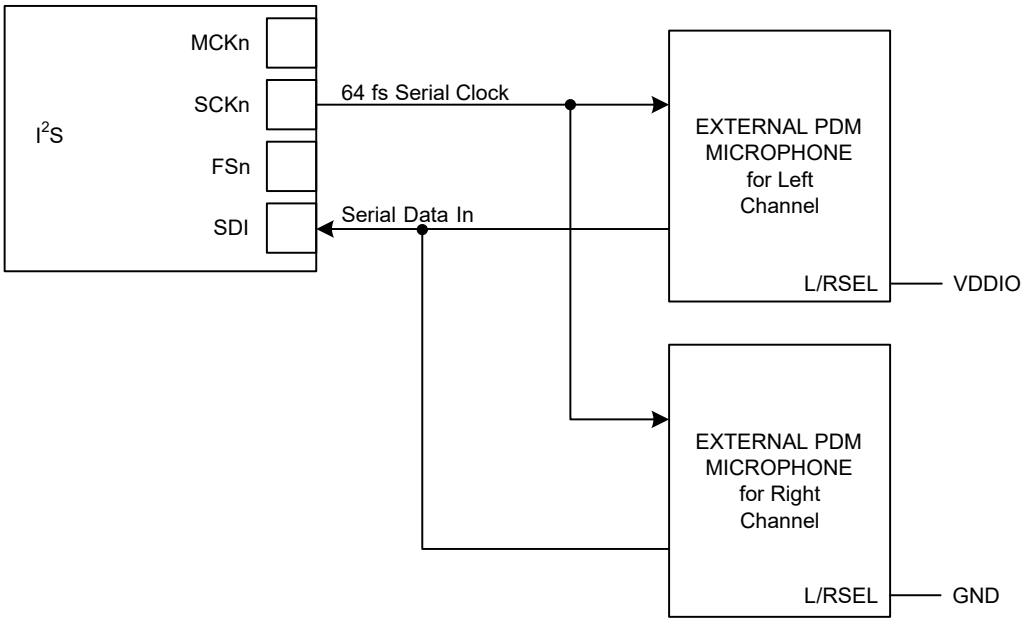

**Figure 51-9.** Codec Application Block Diagram

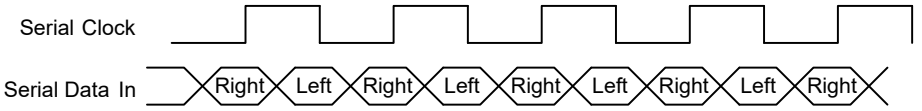




© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 51-10.** PDM Microphones Application Block Diagram





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 51.8. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | RXEN | TXEN | CKEN1 | CKEN0 | ENABLE | SWRST |
| 0x01 ... 0x03 | Reserved | | | | | | | | | |
| 0x04 | CLKCTRL0 | 7:0 | BITDELAY | FSWIDTH[1:0] | | NBSLOTS[2:0] | | | SLOTSIZE[1:0] | |
| | | 15:8 | MCKOUTINV | MCKEN | MCKSEL | SCKOUTINV | SCKSEL | FSOUTINV | FSINV | FSSEL |
| | | 23:16 | | | MCKDIV[5:0] | | | | | |
| | | 31:24 | | | MCKOUTDIV[5:0] | | | | | |
| 0x08 | CLKCTRL1 | 7:0 | BITDELAY | FSWIDTH[1:0] | | NBSLOTS[2:0] | | | SLOTSIZE[1:0] | |
| | | 15:8 | MCKOUTINV | MCKEN | MCKSEL | SCKOUTINV | SCKSEL | FSOUTINV | FSINV | FSSEL |
| | | 23:16 | | | MCKDIV[5:0] | | | | | |
| | | 31:24 | | | MCKOUTDIV[5:0] | | | | | |
| 0x0C | INTENCLR | 7:0 | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| | | 15:8 | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| 0x0E ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | INTENSET | 7:0 | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| | | 15:8 | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| 0x12 ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTFLAG | 7:0 | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| | | 15:8 | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| 0x16 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | SYNCBUSY | 7:0 | | | RXEN | TXEN | CKEN1 | CKEN0 | ENABLE | SWRST |
| | | 15:8 | | | | | | | RXDATA | TXDATA |
| 0x1A ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | TXCTRL | 7:0 | SLOTADJ | | | TXSAME | TXDEFAULT[1:0] | | | |
| | | 15:8 | BITREV | EXTEND[1:0] | | WORDADJ | | DATASIZE[2:0] | | |
| | | 23:16 | SLOTDIS7 | SLOTDIS6 | SLOTDIS5 | SLOTDIS4 | SLOTDIS3 | SLOTDIS2 | SLOTDIS1 | SLOTDIS0 |
| | | 31:24 | | | | | | | DMA | MONO |
| 0x24 | RXCTRL | 7:0 | SLOTADJ | | CLKSEL | | | | SERMODE[1:0] | |
| | | 15:8 | BITREV | EXTEND[1:0] | | WORDADJ | | DATASIZE[2:0] | | |
| | | 23:16 | SLOTDIS7 | SLOTDIS6 | SLOTDIS5 | SLOTDIS4 | SLOTDIS3 | SLOTDIS2 | SLOTDIS1 | SLOTDIS0 |
| | | 31:24 | | | | | | RXLOOP | DMA | MONO |
| 0x28 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | TXDATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x34 | RXDATA | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |

## 51.9. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16-, and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" and/or "Write-Synchronized" property in each individual register description.

Some registers are enable-protected, meaning they can only be written when the module is disabled. Enable protection is denoted by the "Enable-Protected" property in each individual register description.

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 51.9.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | RXEN | TXEN | CKEN1 | CKEN0 | ENABLE | SWRST |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 5 – RXEN** Rx Serializer Enable
Writing a '0' to this bit will disable the Rx Serializer.
Writing a '1' to this bit will enable the Rx Serializer.

| Value | Description |
|---|---|
| 0 | The Rx Serializer is disabled. |
| 1 | The Rx Serializer is enabled. |

**Bit 4 – TXEN** Tx Serializer Enable
Writing a '0' to this bit will disable the Tx Serializer.
Writing a '1' to this bit will enable the Tx Serializer.

| Value | Description |
|---|---|
| 0 | The Tx Serializer is disabled. |
| 1 | The Tx Serializer is enabled. |

**Bits 2, 3 – CKENx** Clock Unit x Enable [x=1..0]
Writing a '0' to this bit will disable the Clock Unit x.
Writing a '1' to this bit will enable the Clock Unit x.

| Value | Description |
|---|---|
| 0 | The Clock Unit x is disabled. |
| 1 | The Clock Unit x is enabled. |

**Bit 1 – ENABLE** Enable
Writing a '0' to this bit will disable the module.
Writing a '1' to this bit will enable the module.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset
Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers to their initial state, and the peripheral will be disabled.
Writing a '1' to CTRL.SWRST will always take precedence, meaning that all other writes in the same write-operation will be discarded.
The $I^2S$ generic clocks must be enabled before triggering Software Reset, hence the logic in all clock domains can be reset.

| Value | Description |
|---|---|
| 0 | There is no reset operation ongoing. |
| 1 | The reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.2. Clock Unit n Control

**Name:** CLKCTRLn
**Offset:** 0x04 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** Enable-Protected, PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | MCKOUTDIV[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | MCKDIV[5:0] | | | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MCKOUTINV | MCKEN | MCKSEL | SCKOUTINV | SCKSEL | FSOUTINV | FSINV | FSSEL |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BITDELAY | FSWIDTH[1:0] | | NBSLOTS[2:0] | | | SLOTSIZE[1:0] | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 29:24 – MCKOUTDIV[5:0]** Host Clock Output Division Factor
The generic clock selected by MCKSEL is divided by (MCKOUTDIV + 1) to obtain the Host Clock n output.

**Bits 21:16 – MCKDIV[5:0]** Host Clock Division Factor
The Host Clock n is divided by (MCKDIV + 1) to obtain the Serial Clock n.

**Bit 15 – MCKOUTINV** Host Clock Output Invert

| Value | Description |
|---|---|
| 0 | The Host Clock n is output without inversion. |
| 1 | The Host Clock n is inverted before being output. |

**Bit 14 – MCKEN** Host Clock Enable
**Note:** MCKEN will not enable the clock output when in Client mode.

| Value | Description |
|---|---|
| 0 | The Host Clock n division and output is disabled. |
| 1 | The Host Clock n division and output is enabled. |

**Bit 13 – MCKSEL** Host Clock Select
This field selects the source of the Host Clock n.

| MCKSEL | Name | Description |
|---|---|---|
| 0x0 | GCLK | GCLK_I2S_n is used as Host Clock n source |
| 0x1 | MCKPIN | MCKn input pin is used as Host Clock n source |

**Bit 12 – SCKOUTINV** Serial Clock Output Invert


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | The Serial Clock n is output without inversion. |
| 1 | The Serial Clock n is inverted before being output. |

**Bit 11 – SCKSEL** Serial Clock Select
This field selects the source of the Serial Clock n.

| SCKSEL | Name | Description |
|---|---|---|
| 0x0 | MCKDIV | Divided Host Clock n is used as Serial Clock n source |
| 0x1 | SCKPIN | SCKn input pin is used as Serial Clock n source |

**Bit 10 – FSOUTINV** Frame Sync Output Invert

| Value | Description |
|---|---|
| 0 | The Frame Sync n is output without inversion. |
| 1 | The Frame Sync n is inverted before being output. |

**Bit 9 – FSINV** Frame Sync Invert

| Value | Description |
|---|---|
| 0 | The Frame Sync n is used without inversion. |
| 1 | The Frame Sync n is inverted before being used. |

**Bit 8 – FSSEL** Frame Sync Select
This field selects the source of the Frame Sync n.

| FSSEL | Name | Description |
|---|---|---|
| 0x0 | SCKDIV | Divided Serial Clock n is used as Frame Sync n source |
| 0x1 | FSPIN | FSn input pin is used as Frame Sync n source |

**Bit 7 – BITDELAY** Data Delay from Frame Sync

| BITDELAY | Name | Description |
|---|---|---|
| 0x0 | LJ | Left Justified (0 Bit Delay) |
| 0x1 | I2S | I2S (1 Bit Delay) |

**Bits 6:5 – FSWIDTH[1:0]** Frame Sync Width
This field selects the duration of the Frame Sync output pulses.
When not in Burst mode, the Clock unit n operates in continuous mode when enabled, with periodic Frame Sync pulses and Data samples.
In Burst mode, a single Data transfer starts at each Frame Sync pulse; these pulses are 1-bit wide and occur only when a Data transfer is requested.
**Note:** The compact stereo modes (16C and 8C) are not supported in the Burst mode.

| FSWIDTH[1:0] | Name | Description |
|---|---|---|
| 0x0 | SLOT | Frame Sync Pulse is 1 Slot wide (default for I2S protocol) |
| 0x1 | HALF | Frame Sync Pulse is half a Frame wide |
| 0x2 | BIT | Frame Sync Pulse is 1 Bit wide |
| 0x3 | BURST | Clock Unit n operates in Burst mode, with a 1-bit wide Frame Sync pulse per Data sample, only when Data transfer is requested |

**Bits 4:2 – NBSLOTS[2:0]** Number of Slots in Frame
Each Frame for Clock Unit n is composed of (NBSLOTS + 1) Slots.

**Bits 1:0 – SLOTSIZE[1:0]** Slot Size
Each Slot for Clock Unit n is composed of a number of bits specified by SLOTSIZE.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| SLOTSIZE[1:0] | Name | Description |
|---|---|---|
| 0x0 | 8 | 8-bit Slot for Clock Unit n |
| 0x1 | 16 | 16-bit Slot for Clock Unit n |
| 0x2 | 24 | 24-bit Slot for Clock Unit n |
| 0x3 | 32 | 32-bit Slot for Clock Unit n |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 51.9.3. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bits 12, 13 – TXURx** Transmit Underrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Underrun x Interrupt Enable bit, which disables the Transmit Underrun x interrupt.

| Value | Description |
|---|---|
| 0 | The Transmit Underrun x interrupt is disabled. |
| 1 | The Transmit Underrun x interrupt is enabled. |

**Bits 8, 9 – TXRDYx** Transmit Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Ready x Interrupt Enable bit, which disables the Transmit Ready x interrupt.

| Value | Description |
|---|---|
| 0 | The Transmit Ready x interrupt is disabled. |
| 1 | The Transmit Ready x interrupt is enabled. |

**Bits 4, 5 – RXORx** Receive Overrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Overrun x Interrupt Enable bit, which disables the Receive Overrun x interrupt.

| Value | Description |
|---|---|
| 0 | The Receive Overrun x interrupt is disabled. |
| 1 | The Receive Overrun x interrupt is enabled. |

**Bits 0, 1 – RXRDYx** Receive Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Ready x Interrupt Enable bit, which disables the Receive Ready x interrupt.

| Value | Description |
|---|---|
| 0 | The Receive Ready x interrupt is disabled. |
| 1 | The Receive Ready x interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.4. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x10
**Reset:** 0x0000
**Property:** PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bits 12, 13 – TXURx** Transmit Underrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Transmit Underrun Interrupt Enable bit, which enables the Transmit Underrun interrupt.

| Value | Description |
|---|---|
| 0 | The Transmit Underrun interrupt is disabled. |
| 1 | The Transmit Underrun interrupt is enabled. |

**Bits 8, 9 – TXRDYx** Transmit Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Transmit Ready Interrupt Enable bit, which enables the Transmit Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Transmit Ready interrupt is disabled. |
| 1 | The Transmit Ready interrupt is enabled. |

**Bits 4, 5 – RXORx** Receive Overrun x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Receive Overrun Interrupt Enable bit, which enables the Receive Overrun interrupt.

| Value | Description |
|---|---|
| 0 | The Receive Overrun interrupt is disabled. |
| 1 | The Receive Overrun interrupt is enabled. |

**Bits 0, 1 – RXRDYx** Receive Ready x Interrupt Enable [x=1..0]
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Receive Ready Interrupt Enable bit, which enables the Receive Ready interrupt.

| Value | Description |
|---|---|
| 0 | The Receive Ready interrupt is disabled. |
| 1 | The Receive Ready interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.5. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x14
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | TXUR1 | TXUR0 | | | TXRDY1 | TXRDY0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | RXOR1 | RXOR0 | | | RXRDY1 | RXRDY0 |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bits 12, 13 – TXURx** Transmit Underrun x [x=1..0]
This flag is cleared by writing a '1' to it.
This flag is set when a Transmit Underrun condition occurs in Sequencer x, and will generate an interrupt request if INTENCLR/SET.TXURx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Underrun x interrupt flag.

**Bits 8, 9 – TXRDYx** Transmit Ready x [x=1..0]
This flag is cleared by writing to DATAx register or writing a '1' to it.
This flag is set when Sequencer x is ready to accept a new data word to be transmitted, and will generate an interrupt request if INTENCLR/SET.TXRDYx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Transmit Ready x interrupt flag.

**Bits 4, 5 – RXORx** Receive Overrun x [x=1..0]
This flag is cleared by writing a '1' to it.
This flag is set when a Receive Overrun condition occurs in Sequencer x, and will generate an interrupt request if INTENCLR/SET.RXORx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Overrun x interrupt flag.

**Bits 0, 1 – RXRDYx** Receive Ready x [x=1..0]
This flag is cleared by reading from DATAx register or writing a '1' to it.
This flag is set when a Sequencer x has received a new data word, and will generate an interrupt request if INTENCLR/SET.RXRDYx is set to '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Receive Ready x interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.6. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x18
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RXDATA | TXDATA |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | RXEN | TXEN | CKEN1 | CKEN0 | ENABLE | SWRST |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 9 – RXDATA** Rx Data Synchronization Status
This bit is cleared when the synchronization of the Rx DATA Holding (RXDATA) register between the clock domains is complete.
This bit is set when the synchronization of the Rx DATA Holding (RXDATA) register between the clock domains is started.

**Bit 8 – TXDATA** Tx Data Synchronization Status
This bit is cleared when the synchronization of the Tx DATA Holding (TXDATA) register between the clock domains is complete.
This bit is set when the synchronization of the Tx DATA Holding (TXDATA) register between the clock domains is started.

**Bit 5 – RXEN** Rx Serializer Enable Synchronization Status
This bit is cleared when the synchronization of the CTRLA.RXEN bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.RXEN bit between the clock domains is started.

**Bit 4 – TXEN** Tx Serializer Enable Synchronization Status
This bit is cleared when the synchronization of the CTRLA.TXEN bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.TXEN bit between the clock domains is started.

**Bits 2, 3 – CKENx** Clock Unit x Enable Synchronization Status [x=1..0]
Bit CKENx is cleared when the synchronization of the CTRLA.CKENx bit between the clock domains is complete.
Bit CKENx is set when the synchronization of the CTRLA.CKENx bit between the clock domains is started.

**Bit 1 – ENABLE** Enable Synchronization Status
This bit is cleared when the synchronization of the CTRLA.ENABLE bit between the clock domains is complete.
This bit is set when the synchronization of the CTRLA.ENABLE bit between the clock domains is started.

**Bit 0 – SWRST** Software Reset Synchronization Status
This bit is cleared when the synchronization of the CTRLA.SWRST bit between the clock domains is complete.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit is set when the synchronization of the CTRLA.SWRST bit between the clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.7. Tx Serializer Control

**Name:** TXCTRL
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** Enable-Protected, Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DMA | MONO |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SLOTDIS7 | SLOTDIS6 | SLOTDIS5 | SLOTDIS4 | SLOTDIS3 | SLOTDIS2 | SLOTDIS1 | SLOTDIS0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BITREV | EXTEND[1:0] | | WORDADJ | | DATASIZE[2:0] | | |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SLOTADJ | | | TXSAME | TXDEFAULT[1:0] | | | |
| Access | R/W | | | R/W | R/W | R/W | | |
| Reset | 0 | | | 0 | 0 | 0 | | |

**Bit 25 – DMA** Single or Multiple DMA Channels
This bit selects whether even-numbered and odd-numbered slots use separate DMA channels or the same DMA channel.

| DMA | Name | Description |
|---|---|---|
| 0x0 | SINGLE | Single DMA channel |
| 0x1 | MULTIPLE | One DMA channel per data channel |

**Bit 24 – MONO** Mono Mode.

| MONO | Name | Description |
|---|---|---|
| 0x0 | STEREO | Normal mode |
| 0x1 | MONO | Left channel data is duplicated to right channel |

**Bits 16, 17, 18, 19, 20, 21, 22, 23 – SLOTDISx** Slot x Disabled for this Serializer [x=7..0]
This field allows disabling some slots in each transmit frame:

| Value | Description |
|---|---|
| 0 | Slot x is used for data transfer. |
| 1 | Slot x is not used for data transfer and will be output as specified in the TXDEFAULT field. |

**Bit 15 – BITREV** Data Formatting Bit Reverse
This bit allows changing the order of data bits in the word in the Formatting Unit.

| BITREV | Name | Description |
|---|---|---|
| 0x0 | MSBIT | Transfer Data Most Significant Bit (MSB) first (default for I2S protocol) |
| 0x1 | LSBIT | Transfer Data Least Significant Bit (LSB) first |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 14:13 – EXTEND[1:0]** Data Formatting Bit Extension
This field defines the bit value used to extend data samples in the Formatting Unit.

| EXTEND[1:0] | Name | Description |
|---|---|---|
| 0x0 | ZERO | Extend with zeros |
| 0x1 | ONE | Extend with ones |
| 0x2 | MSBIT | Extend with Most Significant Bit |
| 0x3 | LSBIT | Extend with Least Significant Bit |

**Bit 12 – WORDADJ** Data Word Formatting Adjust
This field defines left or right adjustment of data samples in the word in the Formatting Unit. for details.

| WORDADJ | Name | Description |
|---|---|---|
| 0x0 | RIGHT | Data is right adjusted in word |
| 0x1 | LEFT | Data is left adjusted in word |

**Bits 10:8 – DATASIZE[2:0]** Data Word Size
This field defines the number of bits in each data sample. For 8-bit compact stereo, two 8-bit data samples are packed in bits 15 to 0 of the DATAm register. For 16-bit compact stereo, two 16-bit data samples are packed in bits 31 to 0 of the DATAm register.

| DATASIZE[2:0] | Name | Description |
|---|---|---|
| 0x0 | 32 | 32 bits |
| 0x1 | 24 | 24 bits |
| 0x2 | 20 | 20 bits |
| 0x3 | 18 | 18 bits |
| 0x4 | 16 | 16 bits |
| 0x5 | 16C | 16 bits compact stereo |
| 0x6 | 8 | 8 bits |
| 0x7 | 8C | 8 bits compact stereo |

**Bit 7 – SLOTADJ** Data Slot Formatting Adjust
This field defines left or right adjustment of data samples in the slot.

| SLOTADJ | Name | Description |
|---|---|---|
| 0x0 | RIGHT | Data is right adjusted in slot |
| 0x1 | LEFT | Data is left adjusted in slot |

**Bit 4 – TXSAME** Transmit Data when Underrun.

| TXSAME | Name | Description |
|---|---|---|
| 0x0 | ZERO | Zero data transmitted in case of underrun |
| 0x1 | SAME | Last data transmitted in case of underrun |

**Bits 3:2 – TXDEFAULT[1:0]** Line Default Line when Slot Disabled
This field defines the default value driven on the SDn output pin during all disabled Slots.

| TXDEFAULT[1:0] | Name | Description |
|---|---|---|
| 0x0 | ZERO | Output Default Value is 0 |
| 0x1 | ONE | Output Default Value is 1 |
| 0x2 | | Reserved |
| 0x3 | HIZ | Output Default Value is high impedance |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.8. Rx Serializer Control

**Name:** RXCTRL
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** Enable-Protected, PAC Write-Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | RXLOOP | DMA | MONO |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SLOTDIS7 | SLOTDIS6 | SLOTDIS5 | SLOTDIS4 | SLOTDIS3 | SLOTDIS2 | SLOTDIS1 | SLOTDIS0 |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BITREV | EXTEND[1:0] | | WORDADJ | | DATASIZE[2:0] | | |
| Access | R/W | R/W | R/W | R/W | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SLOTADJ | | CLKSEL | | | | SERMODE[1:0] | |
| Access | R/W | | R/W | | | | R/W | R/W |
| Reset | 0 | | 0 | | | | 0 | 0 |

**Bit 26 – RXLOOP** Loop-back Test Mode
This bit enables a loop-back test mode:

| Value | Description |
|---|---|
| 0 | Each Receiver uses its SDn pin as input (default mode). |
| 1 | Receiver uses as input the transmitter output of the other Serializer in the pair: e.g. SD1 for SD0 or SD0 for SD1. |

**Bit 25 – DMA** Single or Multiple DMA Channels
This bit selects whether even- and odd-numbered slots use separate DMA channels or the same DMA channel.

| DMA | Name | Description |
|---|---|---|
| 0x0 | SINGLE | Single DMA channel |
| 0x1 | MULTIPLE | One DMA channel per data channel |

**Bit 24 – MONO** Mono Mode.

| MONO | Name | Description |
|---|---|---|
| 0x0 | STEREO | Normal mode |
| 0x1 | MONO | Left channel data is duplicated to right channel |

**Bits 16, 17, 18, 19, 20, 21, 22, 23 – SLOTDISx** Slot x Disabled for this Serializer [x=7..0]
This field allows disabling some slots in each transmit frame:

| Value | Description |
|---|---|
| 0 | Slot x is used for data transfer. |
| 1 | Slot x is not used for data transfer and will be output as specified in the TXDEFAULT field. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 15 – BITREV** Data Formatting Bit Reverse

This bit allows changing the order of data bits in the word in the Formatting Unit.

| BITREV | Name | Description |
|---|---|---|
| 0x0 | MSBIT | Transfer Data Most Significant Bit (MSB) first (default for I2S protocol) |
| 0x1 | LSBIT | Transfer Data Least Significant Bit (LSB) first |

**Bits 14:13 – EXTEND[1:0]** Data Formatting Bit Extension

This field defines the bit value used to extend data samples in the Formatting Unit.

| EXTEND[1:0] | Name | Description |
|---|---|---|
| 0x0 | ZERO | Extend with zeros |
| 0x1 | ONE | Extend with ones |
| 0x2 | MSBIT | Extend with Most Significant Bit |
| 0x3 | LSBIT | Extend with Least Significant Bit |

**Bit 12 – WORDADJ** Data Word Formatting Adjust

This field defines left or right adjustment of data samples in the word in the Formatting Unit. for details.

| WORDADJ | Name | Description |
|---|---|---|
| 0x0 | RIGHT | Data is right adjusted in word |
| 0x1 | LEFT | Data is left adjusted in word |

**Bits 10:8 – DATASIZE[2:0]** Data Word Size

This field defines the number of bits in each data sample. For 8-bit compact stereo, two 8-bit data samples are packed in bits 15 to 0 of the DATAm register. For 16-bit compact stereo, two 16-bit data samples are packed in bits 31 to 0 of the DATAm register.

| DATASIZE[2:0] | Name | Description |
|---|---|---|
| 0x0 | 32 | 32 bits |
| 0x1 | 24 | 24 bits |
| 0x2 | 20 | 20 bits |
| 0x3 | 18 | 18 bits |
| 0x4 | 16 | 16 bits |
| 0x5 | 16C | 16 bits compact stereo |
| 0x6 | 8 | 8 bits |
| 0x7 | 8C | 8 bits compact stereo |

**Bit 7 – SLOTADJ** Data Slot Formatting Adjust

This field defines left or right adjustment of data samples in the slot.

| SLOTADJ | Name | Description |
|---|---|---|
| 0x0 | RIGHT | Data is right adjusted in slot |
| 0x1 | LEFT | Data is left adjusted in slot |

**Bit 5 – CLKSEL** Clock Unit Selection.

| CLKSEL | Name | Description |
|---|---|---|
| 0x0 | CLK0 | Use Clock Unit 0 |
| 0x1 | CLK1 | Use Clock Unit 1 |

**Bits 1:0 – SERMODE[1:0]** Serializer Mode.

| SERMODE[1:0] | Name | Description |
|---|---|---|
| 0x0 | RX | Receive |
| 0x1 | | Reserved |
| 0x2 | PDM2 | Receive one PDM data on each serial clock edge |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**SERMODE** (continued)

| SERMODE[1:0] | Name | Description |
|---|---|---|
| 0x3 | | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.9. Tx Data

**Name:** TXDATA
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Sample Data
This register is used to transfer data to the Tx Serializer.
Data samples written to TXDATA register will be sent to Tx Serializer for transmission, through the Transmit Formatting Unit that will apply the formatting specified in the TXCTRL register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 51.9.10. Rx Data

**Name:** RXDATA
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** Read-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DATA[31:24] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | DATA[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DATA[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DATA[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – DATA[31:0]** Sample Data

This register is used to transfer data from the Rx Serializer.
Data samples received by Rx Serializer will be available for reading from RXDATA register, through the Receive Formatting Unit, according to formatting information for Rx Serializer in the RXCTRL register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 52. PCC - Parallel Capture Controller

## 52.1. Overview

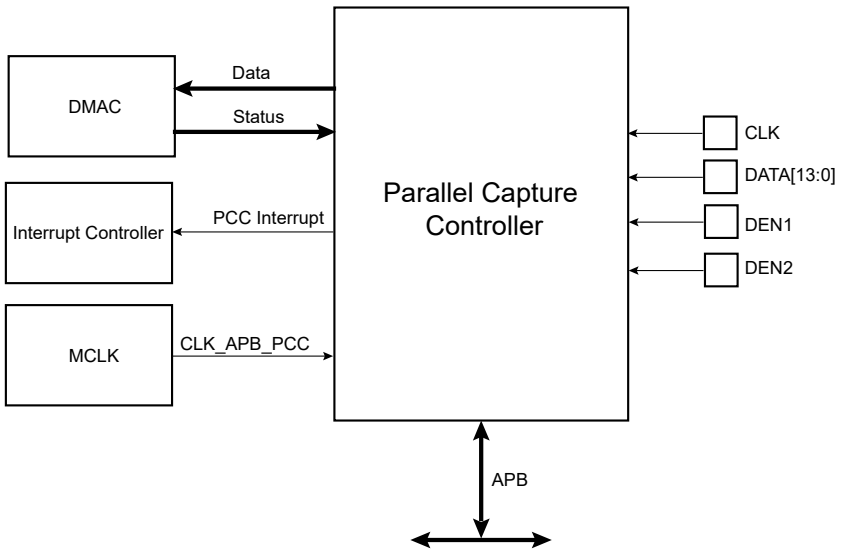
The Parallel Capture Controller can be used to interface an external system, such as a CMOS digital image sensor, ADC, or DSP, and capture its parallel data.

## 52.2. Features

- One clock, up to 14-bit parallel data and two Data Enable on I/O lines
- Data can be sampled every other time (e.g. for chrominance sampling)
- Supports connection of the DMAC which offers buffer reception without processor intervention
- Auto-scale feature available when 10, 12 or 14 bits data size is selected.
- Can be used to interface a CMOS Digital Image Sensor, an ADC, etc.

## 52.3. Block Diagram

**Figure 52-1.** Block Diagram




## 52.4. Signal Description

| Signal | Description | Type |
|---|---|---|
| CLK | Digital input | PCC Clock |
| DATA[13:0] | Digital input | Data [13:0] |
| DEN1 | Digital input | Data Enable 1 |
| DEN2 | Digital input | Data Enable 2 |

## 52.5. Product Dependencies

For the Parallel Capture Controller to function as intended, other interconnected modules of the system must be configured accordingly.

### 52.5.1. I/O Lines

The PCC pins may be multiplexed with the I/O lines Controller. The user must first configure the I/O Controller to assign the PCC pins to their peripheral functions.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.5.2. Power Management

The PCC will continue to operate in any Sleep mode where the selected source clock is running.

### 52.5.3. Clocks

The PCC bus clock (CLK_APB_PCC) is provided by the Main Clock Controller (MCLK) through the AHB-APB D bridge. The clock is enabled and disabled by writing the PCC bit the in the APB D Mask register (MCLK.APBDMASK.PCC). See the register description for the default state of the PCC bus clock.

For capturing operation, the external device has to provide a PCC clock signal (PCC_CLK) synchronous to the data received ("pixel clock") through a pin. See the PORT section and the Multiplexing table for details.

Writing any of the registers does not require the PCC_CLK to be enabled.

**Important:** The CLK_APB_PCC clock frequency must be at least twice the PCC_CLK frequency.

**Related Links**

Register Summary
I/O Multiplexing and Considerations
PORT - I/O Pin Controller

### 52.5.4. DMA

The DMAC can be configured to use the RX channel of the PCC as trigger source.

If configured, a trigger signal is send to the DMAC when data is received by the PCC, such that the DMAC will automatically read the received data buffer. The buffer ready signal will be automatically clear upon the read done by the DMAC.

**Related Links**

Programming Sequence
Without DMAC
With DMAC

### 52.5.5. Interrupts

The PCC has these interrupts:

- OVRE - Overrun Error interrupt
- DRDY - Data Ready interrupt

The interrupt request line is connected to the interrupt controller. Using the interrupts requires the interrupt controller to be configured first. Refer to NVIC - Nested Interrupt Nested Vector Interrupt Controller for details.

### 52.5.6. Events

Not applicable.

### 52.5.7. Debug Operation

When the CPU is halted in debug mode, the PCC will not halt normal operation.
**Note:** A buffer overflow condition will occur if the received data buffer is not read by CPU or CPU DMAC.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.5.8. Register Access Protection

To prevent any single software error from corrupting PCC behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the Write Protection Mode Register (WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the Write Protection Status Register (WPSR) is set and WPSR.WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading WPSR.

The following registers can be write-protected:

- PCC Mode Register

### 52.5.9. Analog Connections

Not applicable.

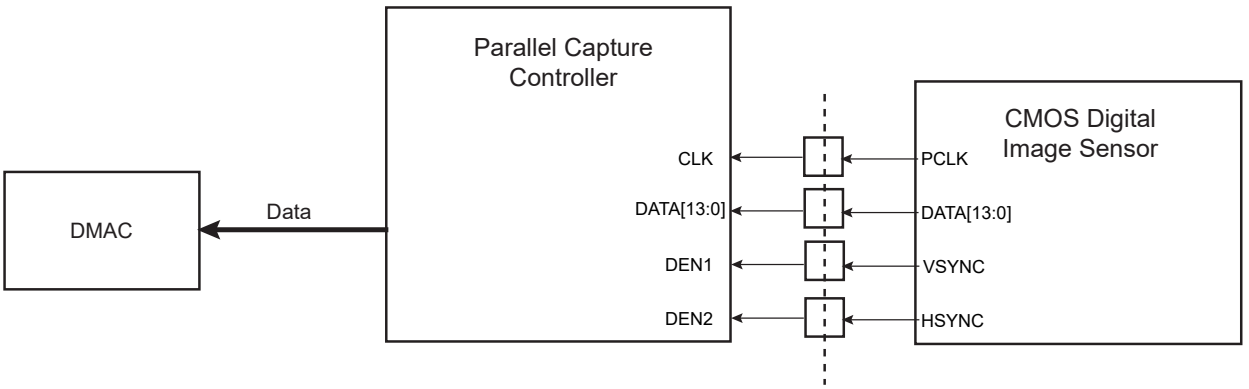
## 52.6. Functional Description

### 52.6.1. Principle of Operation

For better understanding and to ease reading, the following description uses an example with a CMOS digital image sensor.

The CMOS digital image sensor provides a sensor clock, an 10-bit data synchronous with the sensor clock and two data enables which are also synchronous with the sensor clock.

**Figure 52-2.** Parallel Capture Controller Connection with CMOS Digital Image Sensor




The PCC must be configured first, and is enabled by writing a '1' to the Parallel Capture Enable bit in the Mode Register (MR.PCEN).

Once enabled, the PCC samples the data at rising edge of the sensor clock, and resynchronizes it with the PCC clock domain.

The input data bus size can be programmed using the Input Data Size bit field (MR.ISIZE).

A re-initialization of the internal mechanism of the PCC can be automatically done by setting the CID register when a falling edge of the DEN1 or DEN2 is detected. This feature allows glitch filtering and prevents image de-synchronization.

The number of the data which can be read in the Reception Holding Register (RHR) can be programmed by writing the Data Size bit field (MR.DSIZE). The PCC samples one or several sensor data, according to the DSIZE value.

If the MR.SCALE bit is written to '1' and MR.ISIZE ≠ 0, the sampled data is automatically up-scaled to 16 bits. When the right number of data has be sampled, data are stored in the RHR, and the Data Ready flag in the Interrupt Status Register (ISR.DRDY) is set to '1'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The PCC can be associated with a reception channel of the DMA Controller (DMAC). This performs reception transfer from the PCC to a memory buffer without any intervention from the CPU.

The PCC can be configured to either comply with the sensor data enable signals, or not. If the Always Sampling bit in the Mode Register (MR.ALWYS) is written to '0', the PCC samples the sensor data at the rising edge of the sensor clock only if both data enable signals are active (at '1'). If ALWYS is written to '1', the PCC samples the sensor data at the rising edge of the sensor clock, independent of the data enable signals.

The PCC can be configured to sample the sensor data only every other time. This is particularly useful when only the luminance Y from a YUV422 data stream of a CMOS digital image sensor is to be sampled. If the Half Sampling bit in the Mode Register (MR.HALFS) is written to '0', the PCC samples the sensor data as configured above. If MR.HALFS=1, the PCC samples the sensor data as configured above (i.e. respecting the MR.ALWYS setting), but only one time out of two.

The PCC can either sample the even or odd sensor data, depending on the First Sample bit (MR.FRSTS). If sensor data are numbered with an index from zero to n in the order they are received and FRSTS=0, only data with an even index are sampled. For FRSTS=1, only data with an odd index are sampled.

If data are ready in the Reception Holding Register (RHR) but it is not read before new data is stored in RHR, an overrun error occurs: The previous data is lost and the Overrun Error flag in the Interrupt Status Register (ISR.OVRE) is set. This flag is automatically cleared when ISR is read (reset after read).

The flags DRDY and OVRE can be a source of the PCC interrupt.

**Figure 52-3.** PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 0)

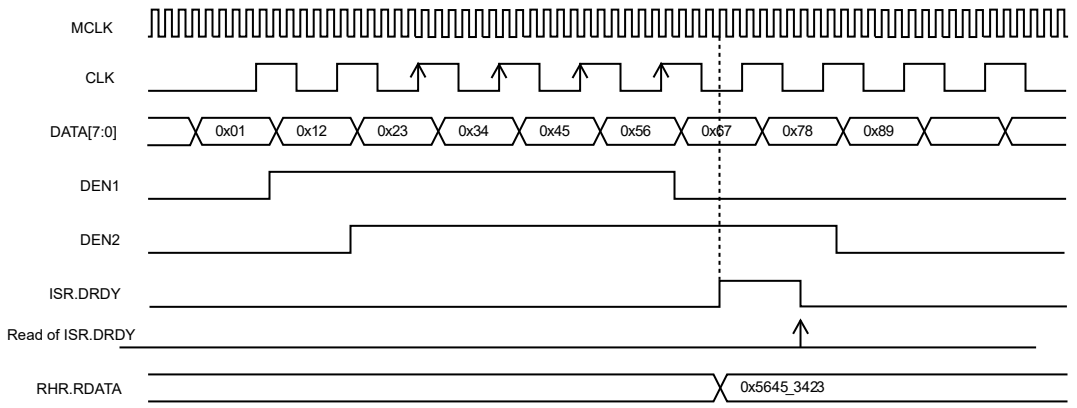
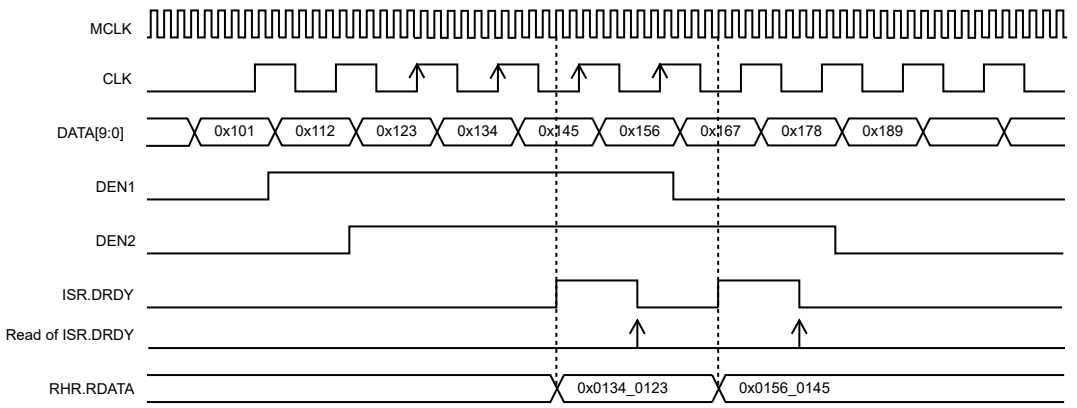



**Figure 52-4.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 0, SCALE = 0)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 52-5.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 0, SCALE = 1)

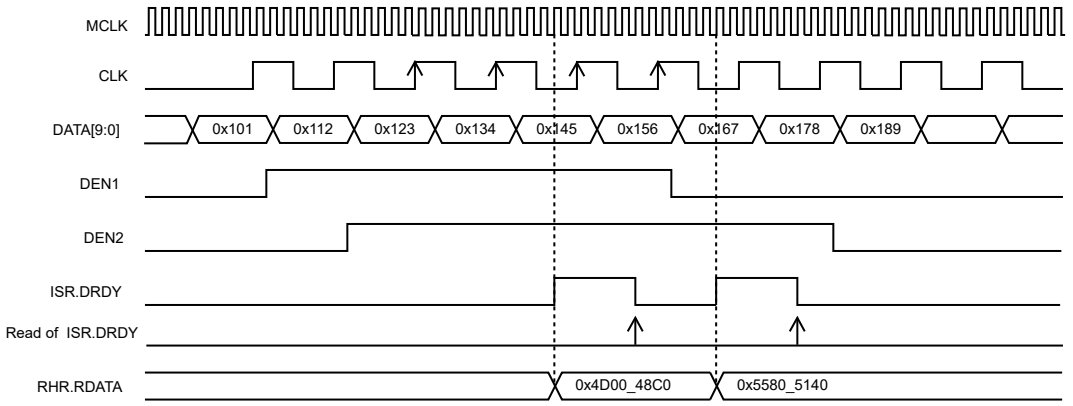



**Figure 52-6.** PCC Waveforms (DSIZE=4_DATA, ALWYS = 1, HALFS = 0)

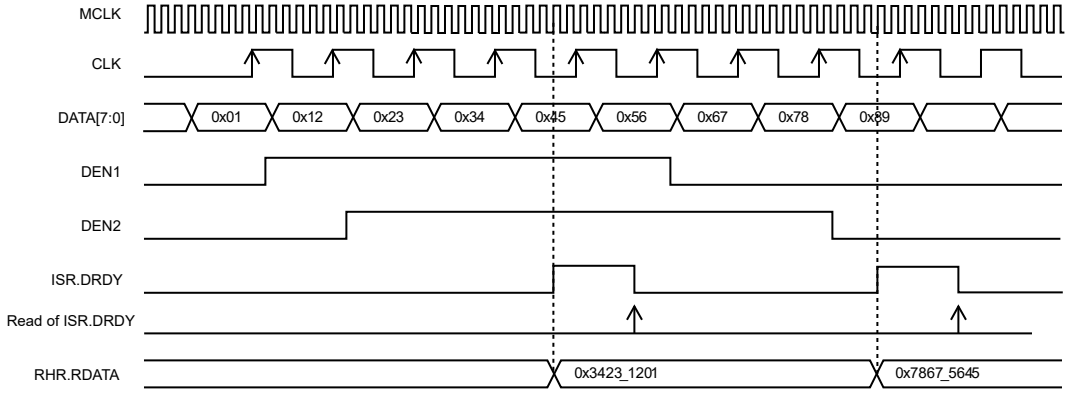
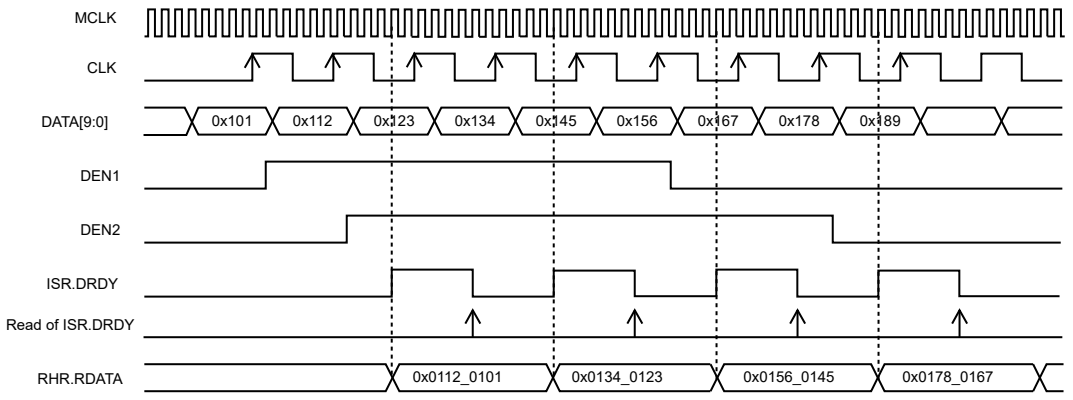



**Figure 52-7.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 1, HALFS = 0, SCALE = 0)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 52-8.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 1, HALFS = 0, SCALE = 1)

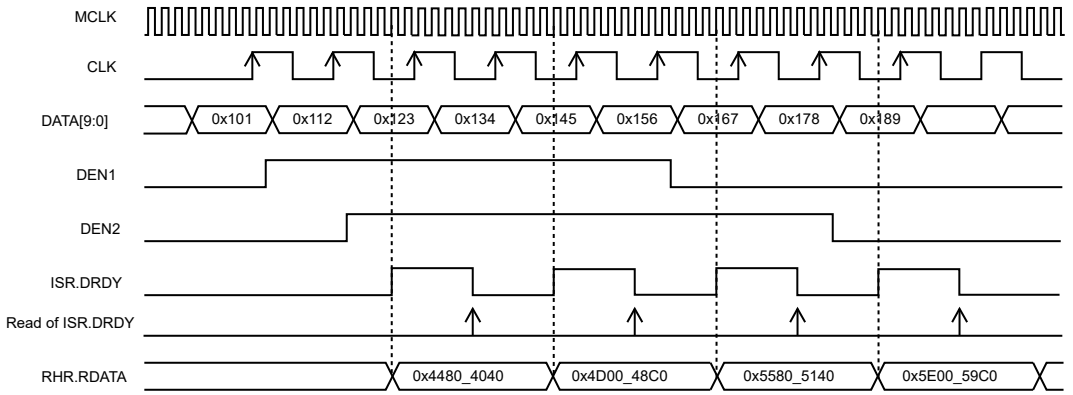



**Figure 52-9.** PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0)

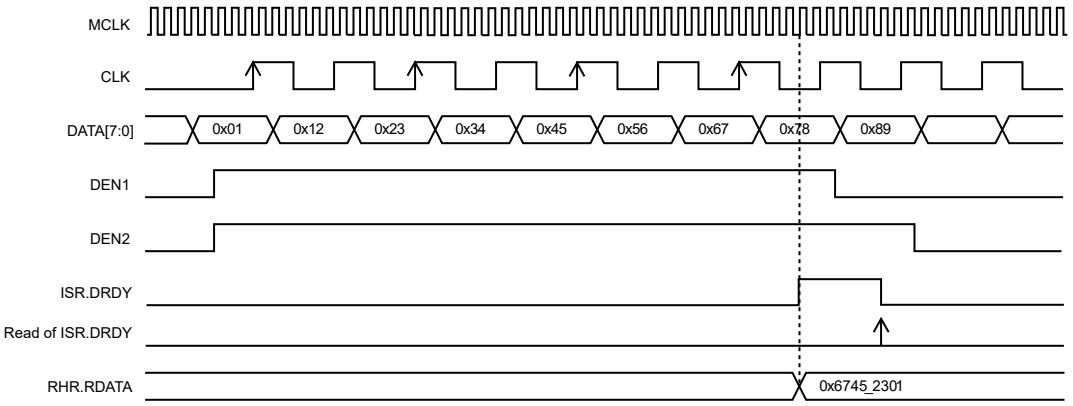
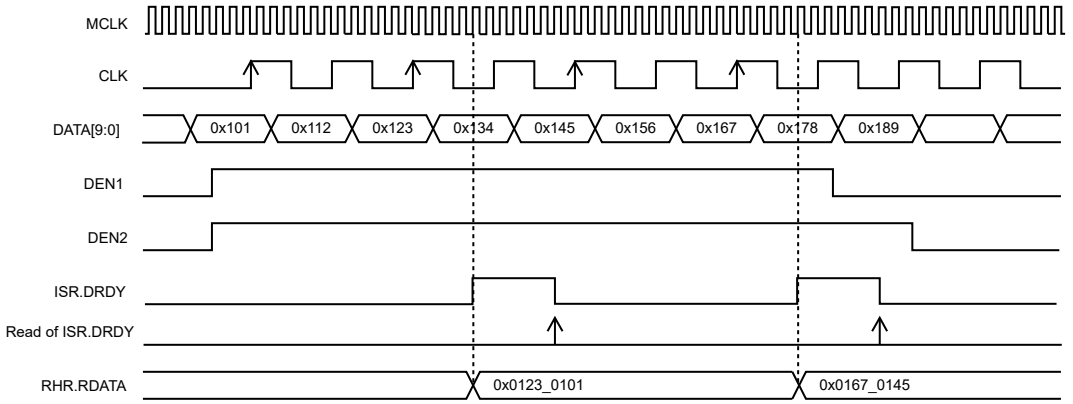



**Figure 52-10.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0, SCALE = 0)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 52-11.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 0, SCALE = 1)

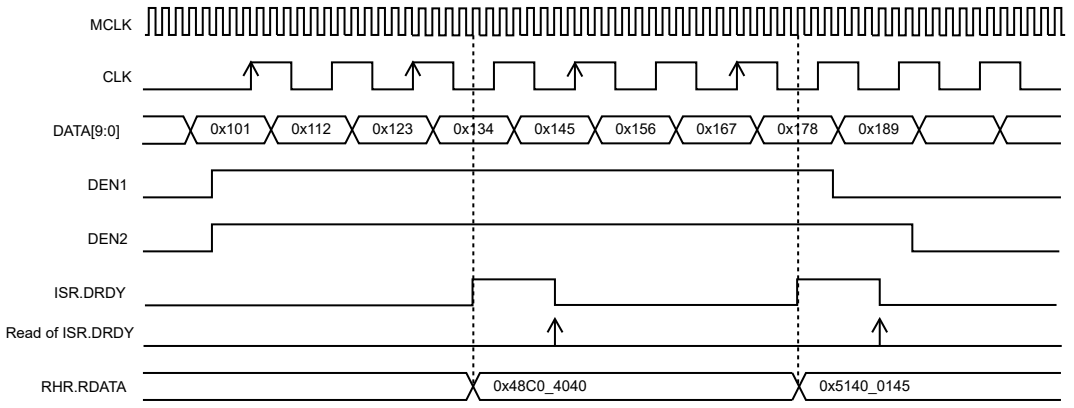



**Figure 52-12.** PCC Waveforms (DSIZE=4_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1)

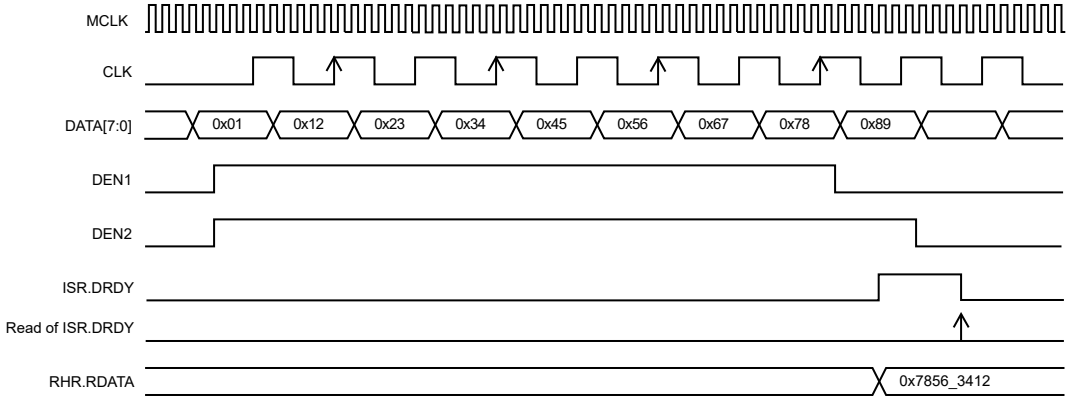
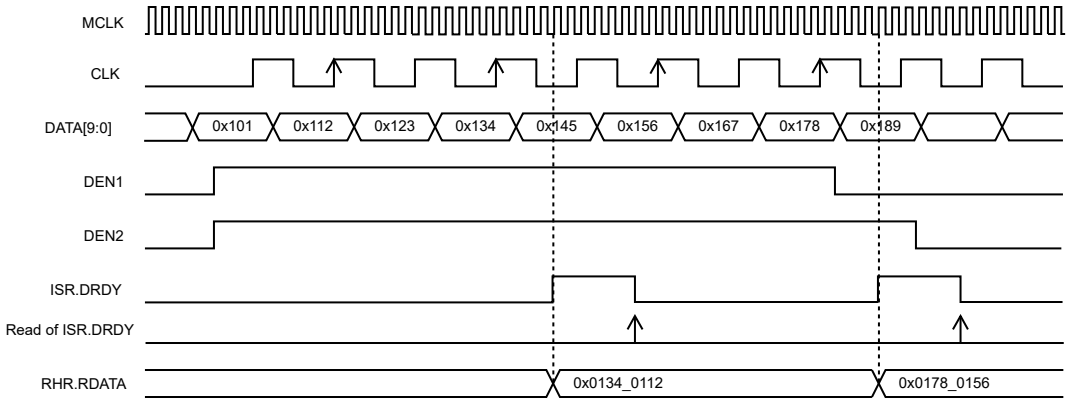


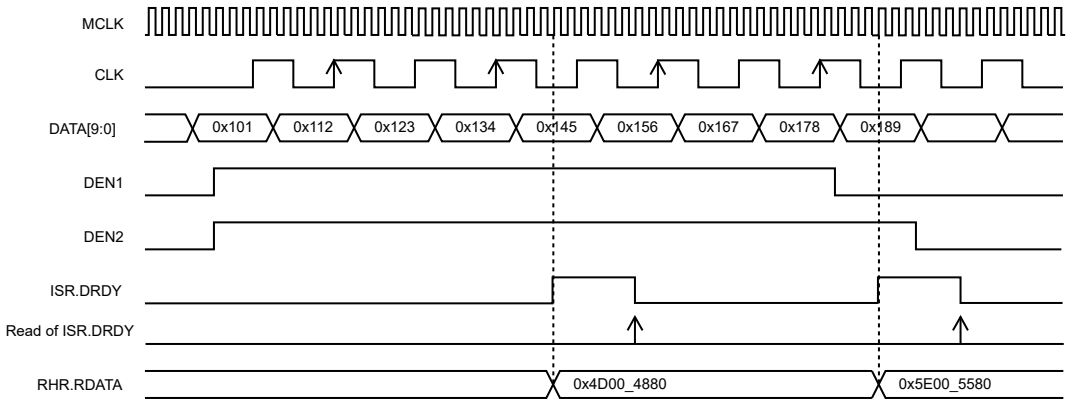

**Figure 52-13.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1, SCALE = 0)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 52-14.** PCC Waveforms (ISIZE=10_BITS, DSIZE=2_DATA, ALWYS = 0, HALFS = 1, FRSTS = 1, SCALE = 1)




### 52.6.2. Register Access Protection

The configuration bit fields ISIZE, SCALE, DSIZE, ALWYS, HALFS and FRSTS in the Mode Register (MR) can be changed ONLY if the PCC is disabled at this time (MR.PCEN=0).

### 52.6.3. Programming Sequence

#### 52.6.3.1. Without DMAC

1. Write the Interrupt Enable and Interrupt Disable Registers (IER and IDR) in order to configure the PCC interrupt mask.
2. Write the Mode Register (MR) fields ISIZE, SCALE, DSIZE, ALWYS, HALFS and FRSTS in order to configure the PCC. Do not enable the PCC in this write access.
3. Write the PCC Enable bit in the Mode Register (MR.PCEN) to '1' in order to enable the PCC. Do not change the configuration from the previous step.
4. Wait for a Data Ready, either by polling the Data Ready flag in the Interrupt Status Register (ISR.DRDY) or by waiting for the corresponding interrupt.
5. Check the Overrun Error flag (ISR.OVRE).
6. Read the data in the Reception Holding Register (RHR).
7. If new data are expected, go to step 4.
8. Disable the PCC by writing MR.PCEN to '0' *without* changing the configuration.

#### 52.6.3.2. With DMAC

1. Write the Interrupt Enable and Interrupt Disable Registers (IER and IDR) in order to configure the PCC interrupt mask.
2. Configure DMAC transfer in the DMAC registers.
3. Write the Mode Register (MR) fields ISIZE, SCALE, DSIZE, ALWYS, HALFS and FRSTS in order to configure the PCC. Do not enable the PCC in this write access.
4. Write the PCC Enable bit in the Mode Register (MR.PCEN) to '1' in order to enable the PCC. Do not change the configuration from the previous step.
5. Wait for the DMA status flag to indicate that the buffer transfer is complete.
6. Check the Overrun Error flag (ISR.OVRE).
7. If a new buffer transfer is expected, go to step 5.
8. Disable the PCC by writing MR.PCEN to '0' *without* changing the configuration.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 52.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | MR | 7:0 | | | DSIZE[1:0] | | | | | PCEN |
| | | 15:8 | | | | | FRSTS | HALFS | ALWYS | SCALE |
| | | 23:16 | | | | | | ISIZE[2:0] | | |
| | | 31:24 | CID[1:0] | | | | | | | |
| 0x04 | IER | 7:0 | | | | | | | OVRE | DRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | IDR | 7:0 | | | | | | | OVRE | DRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | IMR | 7:0 | | | | | | | OVRE | DRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | ISR | 7:0 | | | | | | | OVRE | DRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | RHR | 7:0 | RDATA[7:0] | | | | | | | |
| | | 15:8 | RDATA[15:8] | | | | | | | |
| | | 23:16 | RDATA[23:16] | | | | | | | |
| | | 31:24 | RDATA[31:24] | | | | | | | |
| 0x18 ... 0xDF | Reserved | | | | | | | | | |
| 0xE0 | WPMR | 7:0 | | | | | | | | WPEN |
| | | 15:8 | WPKEY[7:0] | | | | | | | |
| | | 23:16 | WPKEY[15:8] | | | | | | | |
| | | 31:24 | WPKEY[23:16] | | | | | | | |
| 0xE4 | WPSR | 7:0 | | | | | | | | WPVS |
| | | 15:8 | WPVSRC[7:0] | | | | | | | |
| | | 23:16 | WPVSRC[15:8] | | | | | | | |
| | | 31:24 | | | | | | | | |

## 52.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are enable-protected, meaning they can only be written when the peripheral is disabled. Enable-protection is denoted by the "Enable-Protected" property in each individual register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.1. PCC Mode Register

**Name:** MR
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** -

This register can only be written if the WPEN bit is cleared in the Write Protection Mode Register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CID[1:0] | | | | | | | |
| Access | R/W | R/W | | | | | | |
| Reset | 0 | 0 | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ISIZE[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | FRSTS | HALFS | ALWYS | SCALE |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | DSIZE[1:0] | | | | | PCEN |
| Access | | | R/W | R/W | | | | R/W |
| Reset | | | 0 | 0 | | | | 0 |

**Bits 31:30 – CID[1:0]** Clear If Disabled

Clears status flags if disabled. These bits are useful to re-initialize the internal mechanism of the PCC to avoid corrupted data due to glitches. Each time a falling edge of the selected DEN1 or DEN2 signal is detected, the internal mechanism of the PCC is re-initialized to avoid alignment issues.

| Value | Description |
|---|---|
| 0x0 | Clear not enabled |
| 0x1 | Clear on falling edge on DEN1 enabled |
| 0x2 | Clear on falling edge on DEN2 enabled |
| 0x3 | Clear on falling edge on either DEN1 or DEN2 enabled |

**Bits 18:16 – ISIZE[2:0]** Input Data Size

| Value | Name | Description |
|---|---|---|
| 0x0 | 8_BITS | Input data bus size is 8 bits |
| 0x1 | 10_BITS | Input data bus size is 10 bits |
| 0x2 | 12_BITS | Input data bus size is 12 bits |
| 0x3 | 14_BITS | Input data bus size is 14 bits |

**Bit 11 – FRSTS** First Sample

This bit is useful only if the HALFS bit is set to 1. If data are numbered in the order that they are received with an index from 0 to n.

| Value | Description |
|---|---|
| 0 | Only data with an even index are sampled. |
| 1 | Only data with an odd index are sampled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 10 – HALFS** Half Sampling
This function is independent from the ALWYS bit.

| Value | Description |
|---|---|
| 0 | The Parallel Capture Controller samples all the data. |
| 1 | The Parallel Capture Controller samples the data only every other time. |

**Bit 9 – ALWYS** Always Sampling

| Value | Description |
|---|---|
| 0 | The parallel capture Controller samples the data when both data enables are active. |
| 1 | The parallel capture controller always samples the data, regardless of the state of data enable. |

**Bit 8 – SCALE** Scale Data

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | When input data size is not equal to 8 bits (ISIZE ≠ 0), the data stored in the PCC_RHR is automatically up-scaled to 16 bits. |

**Bits 5:4 – DSIZE[1:0]** Data Size

| Value | Name | Description |
|---|---|---|
| 0x0 | 1_DATA | 1 data is read in the PCC_RHR |
| 0x1 | 2_DATA | 2 data are read in the PCC_RHR |
| 0x2 | 4_DATA | 4 data are read in the PCC_RHR (only for 8 bits data size, ISIZE = 0) |
| 0x3 | | Reserved |

**Bit 0 – PCEN** Parallel Capture Enable

| Value | Description |
|---|---|
| 0 | The Parallel Capture Controller is disabled. |
| 1 | The Parallel Capture Controller is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.2. Interrupt Enable Register

**Name:** IER
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | OVRE | DRDY |
| Access | | | | | | | W | W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – OVRE** Overrun Error Interrupt Enable
Writing a '1' to this register enables the Overrun Error interrupt.
Writing a '0' has no effect.

**Bit 0 – DRDY** Data Ready Interrupt Enable
Writing a '1' to this register enables the Data Ready Interrupt interrupt.
Writing a '0' has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.3. Interrupt Disable Register

**Name:** IDR
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | OVRE | DRDY |
| Access | | | | | | | W | W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – OVRE** Overrun Error Interrupt Disable
Writing a '1' to this register disables the Overrun Error interrupt.
Writing a '0' has no effect.

**Bit 0 – DRDY** Data Ready Interrupt Disable
Writing a '1' to this register disables the Data Ready interrupt.
Writing a '0' has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.4. Interrupt Mask Register

**Name:** IMR
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | OVRE | DRDY |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – OVRE** Overrun Error Interrupt Mask

| Value | Description |
|---|---|
| 1 | The Overrun Error interrupt is enabled. |
| 0 | The Overrun Error interrupt is not enabled. |

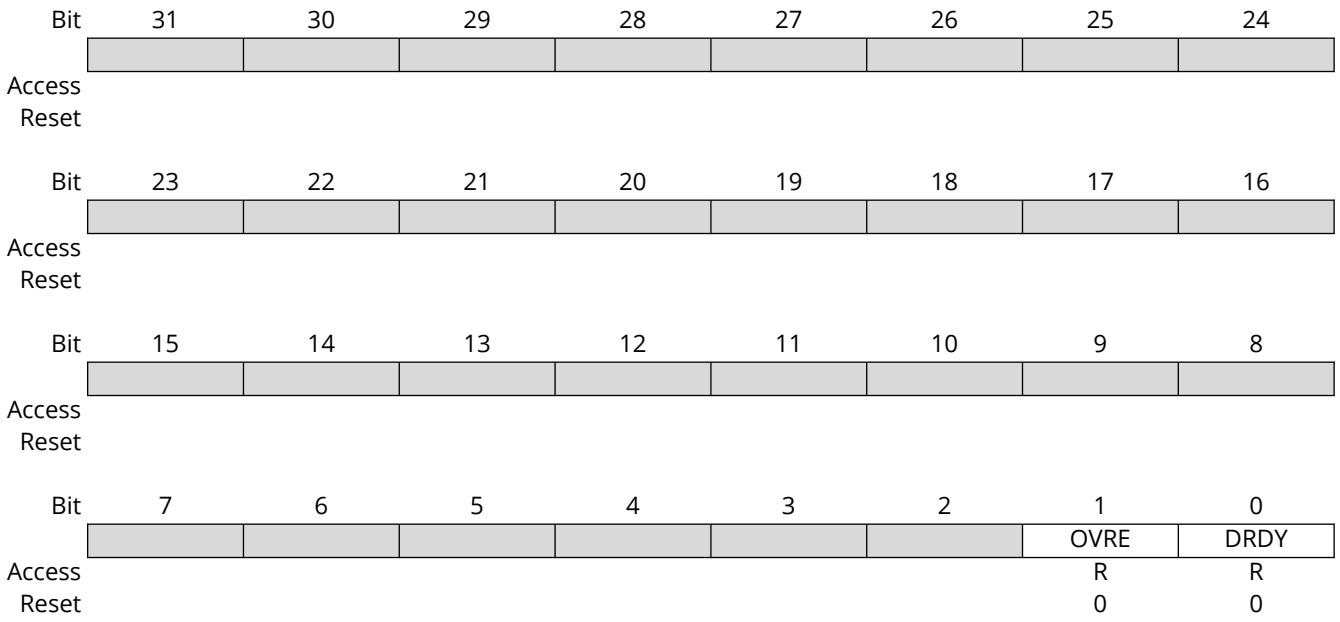
**Bit 0 – DRDY** Data Ready Interrupt Mask

| Value | Description |
|---|---|
| 1 | The Data Ready interrupt is enabled. |
| 0 | The Data Ready interrupt is not enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.5. Interrupt Status Register

**Name:** ISR
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -



| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | OVRE | DRDY |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – OVRE** Overrun Error Interrupt Status
The OVRE flag is automatically reset when this register is read or when the PCC is disabled.

| Value | Description |
|---|---|
| 0 | No overrun error occurred since the last read of this register. |
| 1 | At least one overrun error occurred since the last read of this register. |

**Bit 0 – DRDY** Data Ready Interrupt Status
The DRDY flag is automatically reset when RHR is read or when the PCC is disabled.

| Value | Description |
|---|---|
| 0 | No new data is ready to be read since the last read of RHR. |
| 1 | New data is ready to be read since the last read of RHR. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.6. Reception Holding Register

**Name:** RHR
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RDATA[31:24] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RDATA[23:16] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RDATA[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RDATA[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – RDATA[31:0]** Reception Data

| ISIZE | SCALE | DSIZE | Description |
|---|---|---|---|
| 8_BITS | - | 1_DATA | RDATA[7:0] is useful |
| | | 2_DATA | RDATA[15:0] is useful |
| | | 4_DATA | RDATA[31:0] is useful |
| 10_BITS | 0 (OFF) | 1_DATA | RDATA[9:0] is useful |
| | | 2_DATA | RDATA[9:0] and RDATA[25:16] are useful |
| | 1 (ON) | 1_DATA | RDATA[15:0] is useful |
| | | 2_DATA | RDATA[31:0] is useful |
| 12_BITS | 0 (OFF) | 1_DATA | RDATA[11:0] is useful |
| | | 2_DATA | RDATA[11:0] and RDATA[27:16] are useful |
| | 1 (ON) | 1_DATA | RDATA[15:0] is useful |
| | | 2_DATA | RDATA[31:0] is useful |
| 14_BITS | 0 (OFF) | 1_DATA | RDATA[13:0] is useful |
| | | 2_DATA | RDATA[13:0] and RDATA[29:16] are useful |
| | 1 (ON) | 1_DATA | RDATA[15:0] is useful |
| | | 2_DATA | RDATA[31:0] is useful |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.7. Write Protection Mode Register

**Name:** WPMR
**Offset:** 0xE0
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | WPKEY[23:16] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | WPKEY[15:8] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WPKEY[7:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | WPEN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bits 31:8 – WPKEY[23:0]** Write Protection Key

| Value | Name | Description |
|---|---|---|
| 0x504343 | PASSWD | Writing any other value in this field aborts the write operation of the WPEN bit. Always reads as 0. |

**Bit 0 – WPEN** Write Protection Enable

| Value | Description |
|---|---|
| 0 | Disables the write protection if WPKEY corresponds to 0x504343 (“PCC” in ASCII). |
| 1 | Enables the write protection if WPKEY corresponds to 0x504343 (“PCC” in ASCII). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 52.8.8. Write Protection Status Register

**Name:** WPSR
**Offset:** 0xE4
**Reset:** 0x00000000
**Property:** -

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | WPVSRC[15:8] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WPVSRC[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | WPVS |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

**Bits 23:8 – WPVSRC[15:0]** Write Protection Violation Source
When WPVS = 1, WPVSRC indicates the register address offset at which a write access has been attempted.

**Bit 0 – WPVS** Write Protection Violation Status

| Value | Description |
|---|---|
| 0 | No write protection violation has occurred since the last read of the WPSR. |
| 1 | A write protection violation has occurred since the last read of the WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 53. PDEC – Position Decoder

## 53.1. Overview

The PDEC consists of a Quadrature / Hall decoder, following by a counter, with two compare channels. The counter can be split into two parts to report the angular position and the number of revolutions. If the quadrature decoder feature is not suitable for specific applications, the PDEC module can be used as an additional time base.

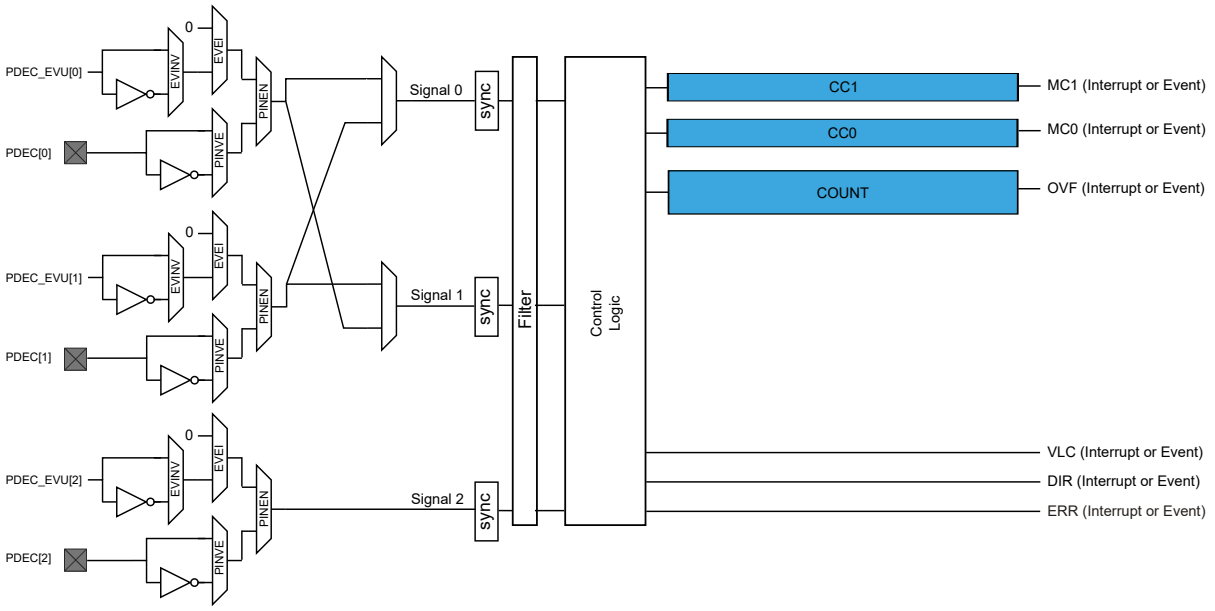
## 53.2. Features

- Internal prescaler
- Selectable mode of operation:
  - QDEC, HALL or COUNTER
- QDEC
  - Angular and revolution counts
  - Synchronous and asynchronous velocity measurements
  - Direction change detection
  - Check valid quadrature transitions
  - Check index position versus angular position
  - Auto correction mode
- HALL
  - Window validation of Hall transitions
  - Hall code detection
  - Direction change detection
  - Check valid Hall transitions
  - Programmable event generation delay after a Hall transition
- COUNTER
  - 16-bit counter with two compare channels
  - One of the compare channels can be configured with period settings
  - Counter overflow interrupt and event generation option
  - Compare match interrupt and event generation option


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 53.3. Block Diagram

**Figure 53-1.** Block Diagram




## 53.4. Signal Description

| Signal Name | Type | Description |
|---|---|---|
| PDEC[2:0] | Digital input | PDEC inputs |

**Note:** One signal can be mapped on one of several pins.

**Related Links**
I/O Multiplexing and Considerations

## 53.5. Product Dependencies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 53.5.1. I/O Lines

Using the I/O lines requires the I/O pins to be configured using the PORT configuration (PORT).

**Related Links**
PORT - I/O Pin Controller

### 53.5.2. Power Management

The PDEC can be configured to operate in any sleep mode. The PDEC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

**Related Links**
PM – Power Manager

### 53.5.3. Clocks


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

A generic clock (GCLK_PDEC) is required to clock the PDEC. This clock must be configured and enabled in the generic clock controller before using the PDEC.

This generic clock is asynchronous to the bus clock (CLK_PDEC_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

**Related Links**
GCLK - Generic Clock Controller
Register Synchronization

### 53.5.4. DMA

Not applicable.

### 53.5.5. Interrupts

The interrupt request line is connected to the Interrupt Controller. In order to use interrupt requests of this peripheral, the Interrupt Controller (NVIC) must be configured first. Refer to *Nested Vector Interrupt Controller* for details.

**Related Links**
Nested Vector Interrupt Controller
Overview
Interrupt Line Mapping

### 53.5.6. Events

The events of this peripheral are connected to the Event System.

**Related Links**
Event System (EVSYS)

### 53.5.7. Debug Operation

When the CPU is halted in debug mode the PDEC will halt normal operation. The PDEC can be forced to continue operation during debugging. Refer to DBGCTRL register for details.

### 53.5.8. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following registers:

- Interrupt Flag register (INTFLAG)
- Filter register (FILTER)
- Precaler register (PRESC)
- Compare x Value register (CCx)
- Channel x Compare Buffer Value register (CCBUFx)
- Status register (STATUS)

Optional write protection by the Peripheral Access Controller (PAC) is denoted by the "PAC Write Protection" property in each individual register description.

PAC write protection does not apply to accesses through an external debugger.

**Related Links**
PAC - Peripheral Access Controller

### 53.5.9. Analog Connections

Not applicable.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 53.6. Functional Description

## 53.6.1. Principle of Operation

The PDEC control logic can be driven by a set of three inputs signal coming from Event System channels or I/O input pins. These three inputs can be filtered prior to down-stream processing. The input polarity, phase definition and other factors are configurable. QDEC, HALL or COUNTER mode of operation are supported.

Depending of the mode configuration, specific input sequences can generate:

- State change
- Counter increment or decrement
- Interrupts
- Output events

## 53.6.2. Basic Operation

### 53.6.2.1. Initialization

The following PDEC registers are enable-protected, meaning they can only be written when the PDEC is disabled (CTRLA.ENABLE is zero):

- Event Control register (EVCTRL)

Enable-protected bits in the CTRLA register can be written at the same time as CTRLA.ENABLE is written to '1', but not at the same time as CTRLA.ENABLE is written to '0'.

Enable-protection is denoted by the 'Enable-Protected' property in the register description.

### 53.6.2.2. Enabling, Disabling, and Resetting

The PDEC must be configured before it is enabled by the following steps:

1. Enable the PDEC bus clock (CLK_PDEC_APB)
2. Select the mode of operation by writing the Mode bits in the Control A register (CTRLA.MODE)
3. Select the PDEC mode configuration by writing the Configuration bits in the Control A register (CTRLA.CONF)
4. Select the PDEC event or pin input signal source by writing the Event Enable Input bit in the Event Control register (EVCTRL.EVEI) or by the Pin Enable bit in Control A register (CTRLA.PINEN)
5. Select the angular counter length value by writing the Angular bits in the Control A register (CTRLA.ANGULAR)

Optionally, the following configurations can be set before enabling PDEC:

- The GCLK_PDEC clock can be prescaled by writing to the Prescaler register (PRESC)
- A filter can be applied to the input signal by writing a corresponding value to the Filter register (FILTER)
- If the resolution of the rotary sensor is not a power of 2, an Angular period can be set (CTRLA.PEREN and CC0 register)

The PDEC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The PDEC is disabled by writing a '0' to CTRLA.ENABLE.

In QDEC or HALL operation modes, PDEC decoding is enabled writing a START command in the Control B Set register (CTRLBSET.CMD=START). The PDEC decoding is disabled writing a STOP command in the Control B Set register (CTRLBSET.CMD=STOP).

The PDEC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the PDEC, except DBGCTRL, will be reset to their initial state, and the PDEC will be disabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

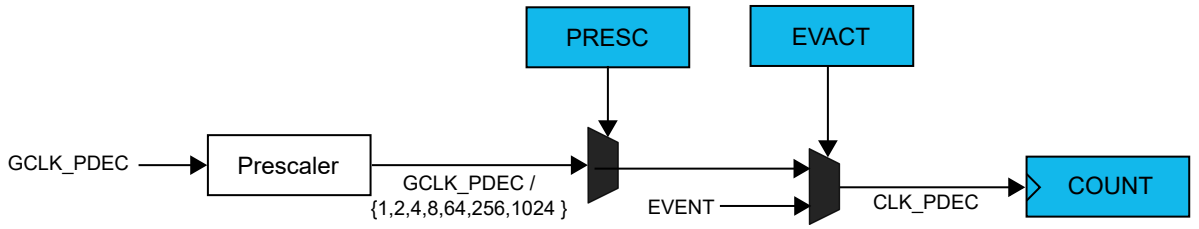
The PDEC should be disabled before the PDEC is reset to avoid undefined behavior.

### 53.6.2.3. Prescaler Selection

The GCLK_PDEC is fed into the internal prescaler. Prescaler outputs from 1 to 1/1024 are directly available for selection by the counter and all selections are available in Prescaler register (PRESC). If the prescaler value is higher than 0x01, the counter update condition is executed on the next prescaled clock pulse.

If the counter is set to count events, the internal prescaler is bypassed and the GCLK_PDEC clock is automatically selected during operation. The prescaler clock is also enabled when the input filtering is required.

**Figure 53-2.** Prescaler Selection



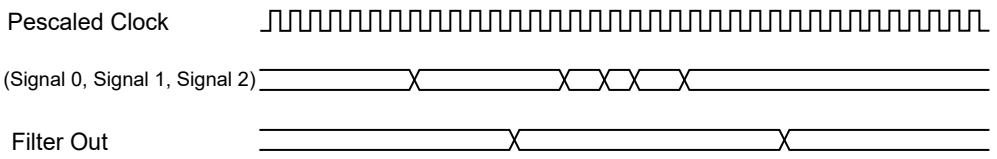

### 53.6.2.4. Input Selection and Filtering

The QDEC and HALL operations require three inputs, as illustrated in the figure below.. Each input can either be a dedicated I/O pin or an Event system channel. This is selected by writing to the corresponding Event x Enable bit in the Event Control register (EVCTRL.EVEIx) or Pin x Enable bit in the Control A register (CTRLA.PINENx).

The I/O input pin active level can be inverted by writing to the corresponding Pin x Inversion Enable bit in the Control A register (CTRLA.PINVENx). Similarly, the event input active level can be inverted by writing to the corresponding Inverted Event x Input Enable bit in Event Control register (EVCTRL.EVINVx).

All input signals can be filtered before they are fed into the control logic. The FILTER register is used to configure the minimum duration for which the input signal has to be valid. The input signal minimum duration must be (FILTER+1)* $t_{GCLK_PDEC}$ .

**Figure 53-3.** Input Signal Filtering




Only the first two input signals can be swapped by writing to the SWAP bit in the Control A register (CTRLA.SWAP).

**Related Links**

Block Diagram

### 53.6.2.5. Period Control

The Channel Compare 0 register (CC0) can act as a period register (PER) by writing the PEREN bit in the Control A register (CTRLA.PEREN) to '1'. The PER can be used to control the top value (TOP) of the counting operation:

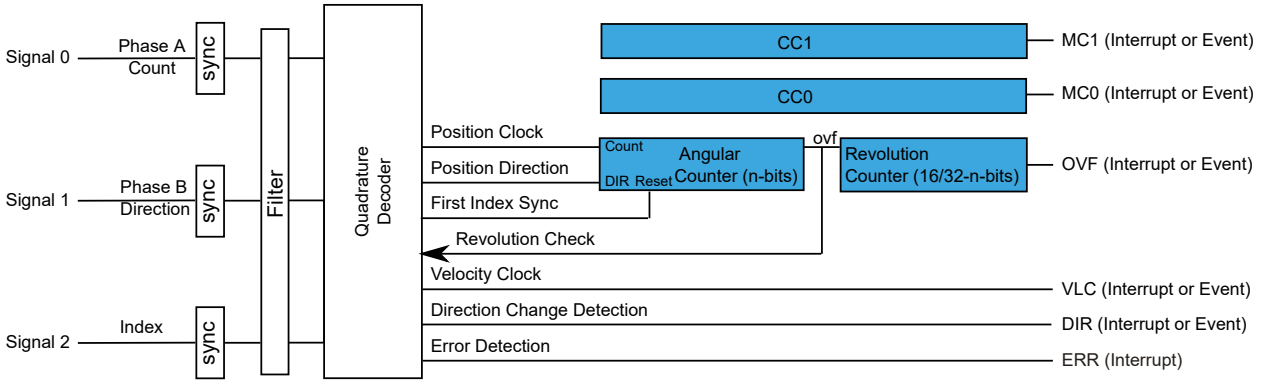

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When up-counting and the counter reaches the value of CC0, the counter is cleared to zero. When down-counting and the counter reaches zero, the counter is reloaded with the CC0 value.

### 53.6.2.6. QDEC Operation Mode

In QDEC mode of operation, Signal 0 and Signal 1 control logic inputs refer to Phase A and Phase B in X4 mode, and to count/direction in X2 mode. The Signal 2 control logic input refers to the Index, in both X4 and X2 mode of operation. In X4 mode, a simultaneous transition on Phase A and Phase B will cause a QDEC error detection (STATUS.QERR).

**Figure 53-4.** QDEC Block Diagram




**Related Links**

Block Diagram

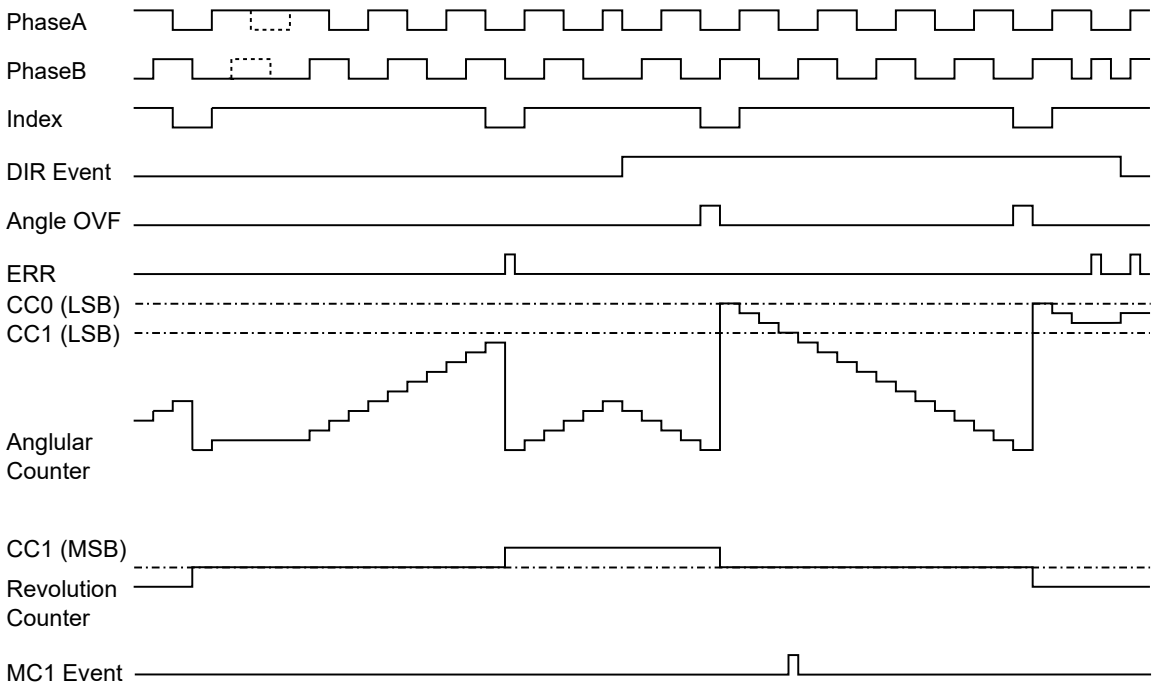
#### 53.6.2.6.1. Position and Rotation Measurement

After filtering, the quadrature signals are analyzed to extract the rotation direction and edges in order to be counted by the counter.

The counter is split in two parts, Angular and Revolution. The Phase A and B edge detections define the motor axis position, which is recorded by the Angular part of the counter. The motor revolution is recorded by the Revolution part of the counter. The Angular counter is updated each time a QDEC transition is detected. The Revolution counter is updated on each angular counter overflow or underflow.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 53-5.** Position and Rotation Measurement




in Q4 and Q4S configuration, a valid index is detected when the three inputs (PhaseA, PhaseB and Index) are at low level.

In Q2 and Q2S configuration, a valid index is detected when the two inputs (Count and Index) are at low level.

in Q2 and Q4 configuration, depending on current detected direction, Index will reset or reload the Angular counter and increment or decrement the Revolution counter.

In Q2S and Q4S configuration, the Angular counter is reset on the first Index occurrence after the PDEC decoding is enabled. When any next Index occurrence does not match an Angular counter overflow or underflow, the Index Error flag in Status register is set (STATUS.IDXERR). The Error Interrupt Flag is set (INTFLAG.ERR) and an optional interrupt can be generated.

An Index Error is also generated after the PDEC decoding is enabled and no Index has been detected after one Angular counter revolution.

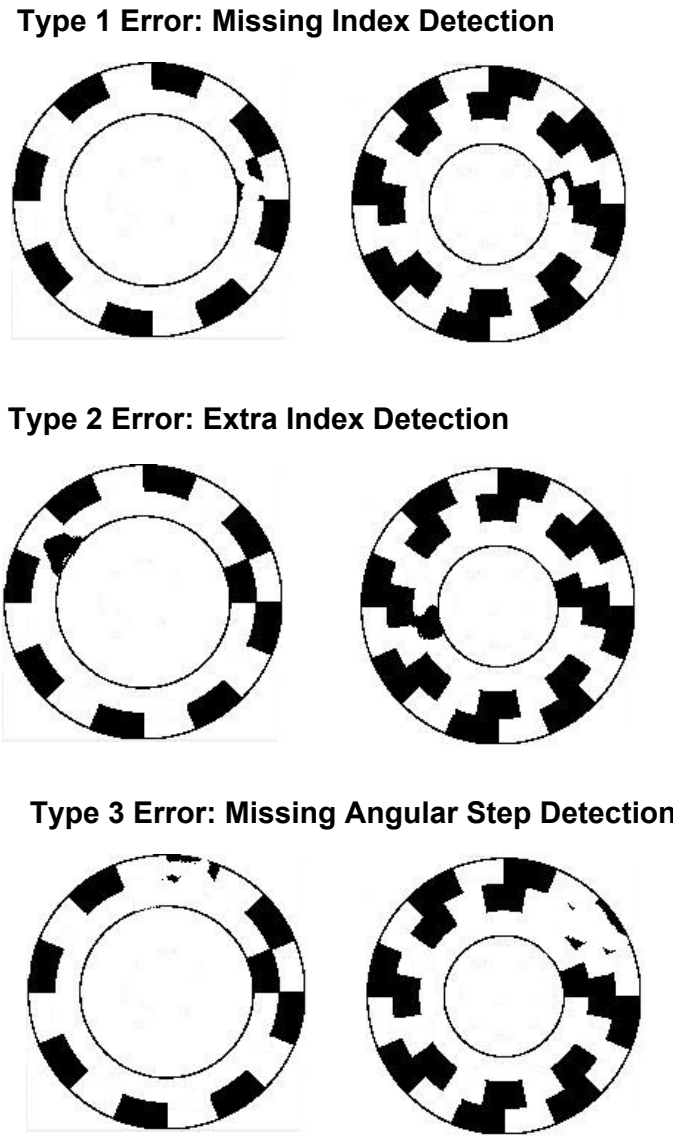
#### 53.6.2.6.2. Secure Decoder Detection

Position decoders are generally implemented using external photo-detectors, detecting a transmitted or reflected beam. Error detections or no detections may happen, and are due to:

- A transmitted bean locally stopped by a dust
- A reflected beam lost by a mat dust
- An additional parasitic reflected beam, due to a gloss metallic dust


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 53-6.** Coding Wheels Error Types

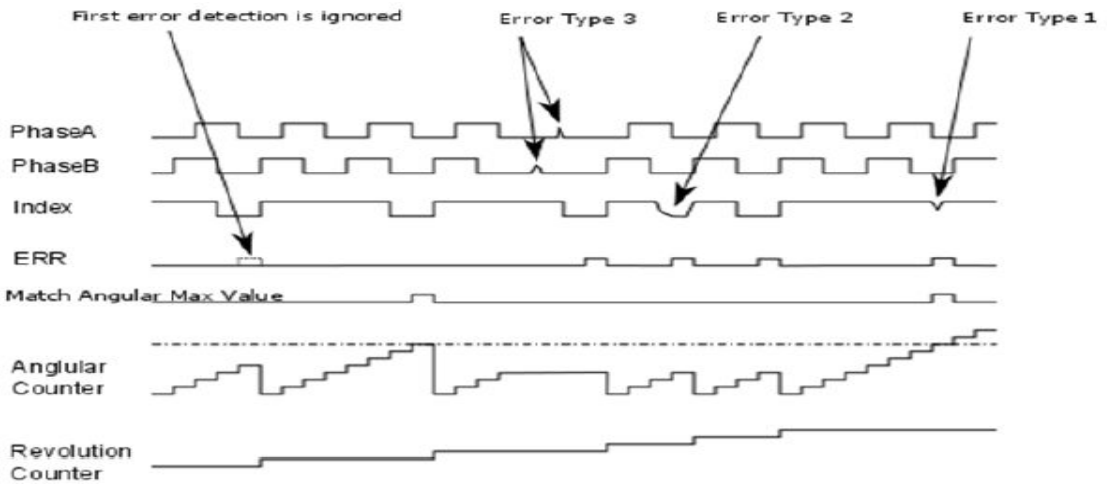



When secure detection is enabled (CTRLA.CONF = X4S or CTRLA.CONF = X2S), the Index must be enabled. The Angular counter is restarted on the detection of each overflow. When the Angular counter value reaches its maximum period value (TOP), an Index is expected to be detected. In the same way, if the Index is detected, the Angular counter value is expected to be TOP. If one of these conditions is not met, an error is generated.

**Note:** The first error generation is masked, as the initial position of the wheel is in unknown state. The first index is used to synchronize the Angular counter on Revolution counter increment.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 53-7.** Secure Decoder Detections

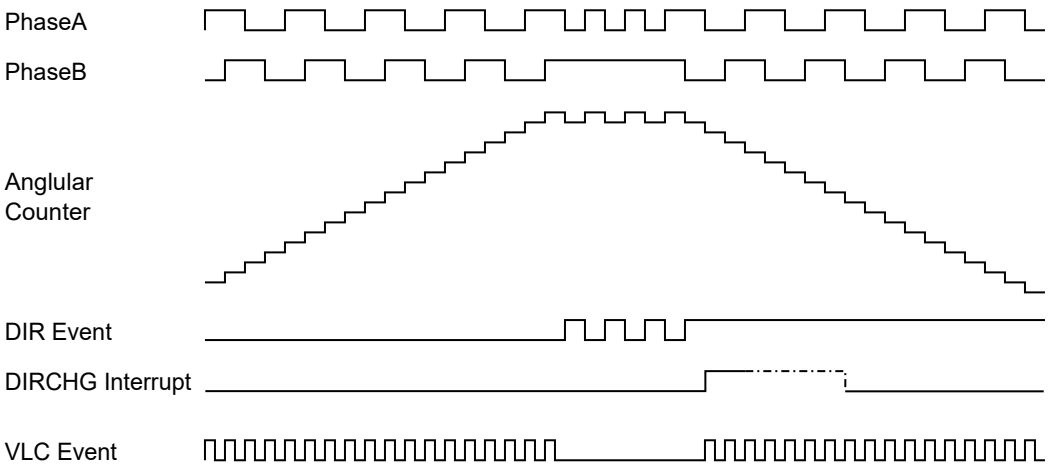



### 53.6.2.6.3. Direction Status and Change Detection

The direction (DIR) status can be directly read anytime in the STATUS register (STATUS.DIR). The polarity of the direction flag status depends of the input signal swap and active level configuration.

Each time a rotation direction change is detected, the Direction Change Interrupt Flag is set (INTFLAG.DIR) and an optional interrupt can be generated. The same interrupt condition is source of Direction event output.

**Figure 53-8.** Rotation Direction Change




To avoid spurious interrupts when coding wheel is stopped, the direction change condition is reported as an interrupt, only on the second edge confirming the direction change.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

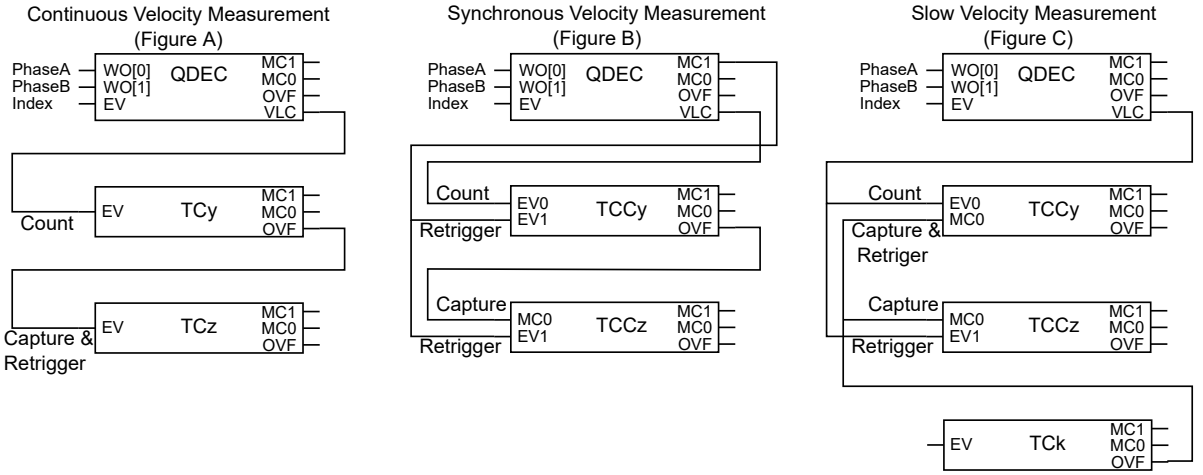
Velocity output event is generated on each QDEC transition except when the direction changes.

#### 53.6.2.6.4. Speed Measurement

Three types of speed measurement can be done using velocity event output (VLC) and Timer/Counter (TC/TCC) device resources.

- Continuous velocity measurement: TCz measures the time on which n VLC (TCy) output events occur
- Synchronous Velocity measurement: On a specific motor position TCCz, the time is measured on which n VLC (TCCy) output events occur.
- Slow Velocity measurement: measure the number of VLC output events (TCCy) plus the delay since the last VLC output event (TCCz) within a given time slot (TCk).

**Figure 53-9.** Speed Measurement




#### 53.6.2.6.5. Missing Pulse Detection and Auto-Correction

The PDEC embeds circuitry to detect and correct errors that may result from contamination on optical disks or other sources producing quadrature phase signals.

The auto-correction works in QDEC X4S mode only. A missing pulse on a phase signal is automatically detected, and the pulse count reported in the Angular part of COUNT is automatically corrected.

There is no autocorrection if both phase signals are affected at the same location on the input signals, because the autocorrection requires a valid phase signal to detect contamination on the other phase signal.

If the quadrature source is undamaged, the number of pulses counted for a predefined period of time must be the same with or without detection and auto-correction. Therefore, if the measurement results differ, a contamination exists on the source producing the quadrature signals. This does not substitute the measurements of the number of pulses between two index pulses (if available) but provides an additional method to detect damaged quadrature sources.

When the source providing quadrature signals is strongly damaged, potentially leading to a number of consecutive missing pulses greater than 1, the quadrature decoder processing may be affected.

The Maximum Consecutive Missing Pulses bits in Control A register (CTRLA.MAXCMP) define the maximum acceptable number of consecutive missing pulses. If the limit is reached, the Missing Pulse Error flag in Status register (STATUS.MPERR) is set. The Error Interrupt flag is set (INTFLAG.ERR) and an optional interrupt can be generated.

**Note:** When the MAXCMP value is zero, the MPERR error flag is never set.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

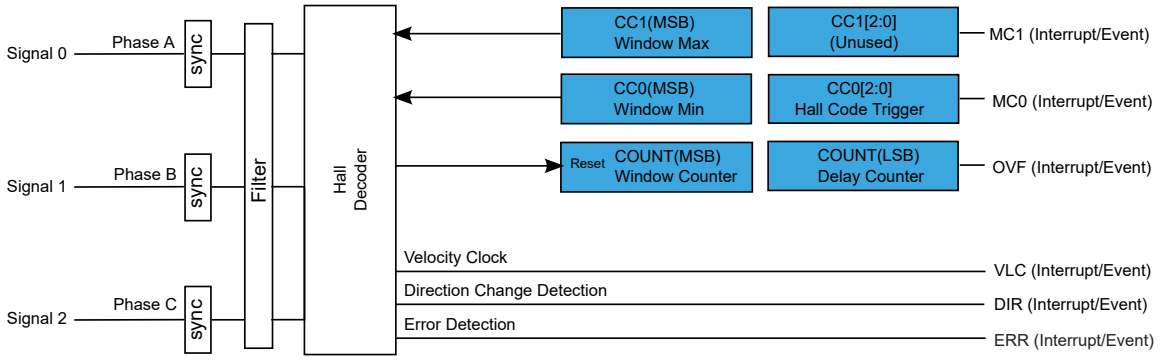
## 53.6.3. Additional Features

### 53.6.3.1. HALL Operation Mode

In HALL operation mode, control logic signal 0, 1 and 2 inputs represent the phase A, B and C of a Hall sensor, respectively.

A programmable delayed event can be generated to update a TCC pattern generator.

**Figure 53-10.** HALL Block Diagram




**Related Links**

Block Diagram

#### 53.6.3.1.1. Hall Sensor Control

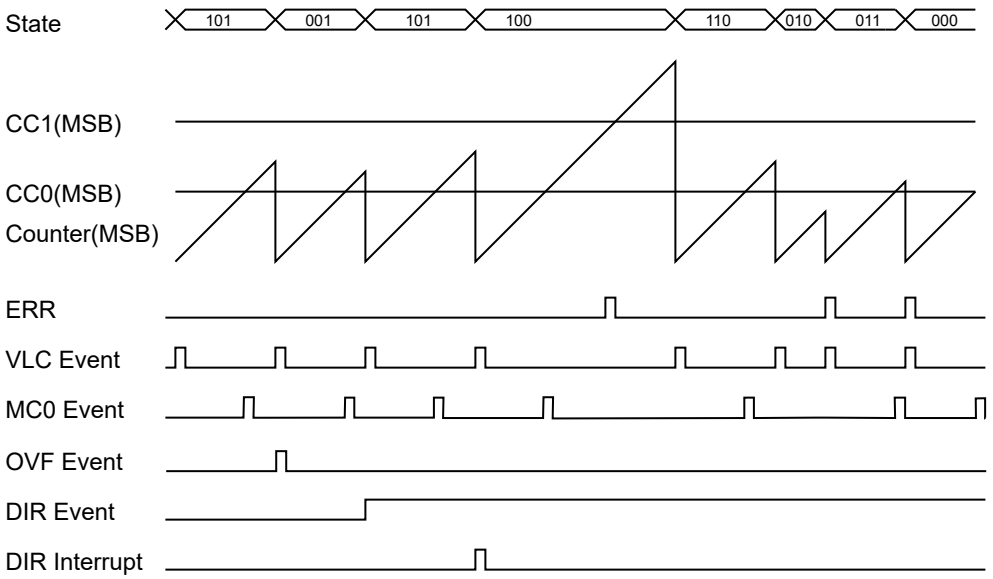
On any update of the filter output:

- The filter output value is checked to be a valid Hall value. If an invalid Hall code is reported, the Hall Error bit in Status register will be set (STATUS.HERR).
- The MC0 Interrupt Flag bit is set (INTFLAG.MC0) if CC0[2:0] matches the filter output value. An optional compare match interrupt or Event output is generated on the same condition detection.
- The window counter is checked to be between CC0[MSB] and CC1[MSB] value, and reset to 0 value. If an error is detected, the Window Error bit in Status register (STATUS.WINERR) is set.
- The delay counter is started, and MC0 optional interrupt or event is generated when the delay counter matches CC0[LSB].

Any error condition will set the Error Interrupt Flag (INTFLAG.ERR). An optional interrupt or event output is generated on the same condition detection.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 53-11.** Hall Waveforms




### 53.6.3.2. Counter Operation Mode

Depending on the mode of operation, the counter (Counter Value register COUNT) is cleared, reloaded, or incremented at each counter clock input.

The counter will count for each clock tick until it reaches TOP. When TOP is reached, the counter will be set to zero on the next clock input.

This comparison will set the Overflow Interrupt Flag in the Interrupt Flag Status and Clear register (INTFLAG.OVF) and can be used to trigger an interrupt or an event.

It is possible to change the counter value when the counter is running. The write access has higher priority than count, or clear. The COUNT value will always be zero when starting the PDEC, unless a different value has been written to it, or the PDEC has been disabled at a value other than zero. Due to asynchronous clock domains, the internal counter settings are written once the synchronization is complete.

**Related Links**

GCLK - Generic Clock Controller

### 53.6.3.3. Register Lock Update

Prescaler (PRESC), FILTER, and CCx registers are buffered (PRESCBUF, FILTERBUF, CCBUFx registers, respectively). When a new value is written in a buffer register, the corresponding Buffer Valid bit is set in the Buffer Status register (STATUS.FILTERBUFV, STATUS.PRESCBUFV, STATUS.CCBUFVx).

By default, a register is updated with the its buffer register's value on UPDATE condition, which represents:

- The next filter transition in QDEC and HALL mode of operation
- The overflow/underflow or re-trigger event detection in COUNT mode of operation

The buffer valid flags in the STATUS register are automatically cleared by hardware when the data is copied from the buffer to the corresponding register.

It is possible to lock the updates by writing a '1' to the Lock Update bit in Control B Set register (CTRLBSET.LUPD).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The lock feature is disabled by writing a '1' to the Lock Update bit in Control B Clear register (CTRLBCLR.LUPD). When a buffer valid status flag is '1' and updating is not locked, the data from the buffer register will be copied into the corresponding register on UPDATE condition.

It is also possible to modify the LUPD bit behavior by hardware, by writing a '1' to the Auto-lock bit in Control A register (CTRLA.ALOCK). When the bit is '1', the Lock Update bit in Control B register (CTRLBSET.LUPD) is set when the UPDATE condition is detected.

#### 53.6.3.4. Software Command and Event Actions

The PDEC peripheral supports software commands and event actions. The software commands are applied by the Software Command bit field in the Control B register (CTRLBSET.CMD, CTRLBCLR.CMD). The event actions are available in the Event Action bit-field in Event Control register (EVCTRL.EVACT).

##### 53.6.3.4.1. Re-trigger Software Command or Event Action

A re-trigger command can be issued from software by using PDEC Command bits in Control B Set register (CTRLBSET.CMD = RETRIGGER) or when the re-trigger event action is configured in the Input Event Action bits in Event Control register (EVCTRL.EVACT = RETRIGGER) and an event is detected by hardware.

When the re-trigger command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (DIR). If the re-trigger command is detected when the counter is stopped, the counter will resume counting operation from the value in the COUNT register.

**Note:** When re-trigger event action is enabled, enabling the counter will not start the counter. The counter will start on the next incoming event and restart on any following event.

##### 53.6.3.4.2. Count Event Action

The count action can be selected in the Event Control register (EVCTRL.EVACT) and can be used to count external events. When an event is received, the counter increments the value.

##### 53.6.3.4.3. Force Update Software Command

A Force Update command can be issued by writing the PDEC Command bits in Control B Set register (CTRLBSET.CMD = UPDATE). When the command is issued, the buffered registers will be updated.

##### 53.6.3.4.4. Force Read Synchronization Software Command

A Force Read Synchronization command can be issued writing the PDEC Command bits in Control B Set register (CTRLBSET.CMD = READSYNC). When the command is issued, a COUNT register read synchronization is forced.
**Note:** This command should be used to read the most updated COUNT internal value.

### 53.6.4. Interrupts

The PDEC has the following interrupt sources:

- Overflow/Underflow: OVF
- Compare Channels: COMPx
- Error: ERR
- Velocity: VLC. This interrupt is available only in QDEC and HALL operation modes.
- Direction: DIR. This interrupt is available only in QDEC and HALL operation modes.

Each interrupt source has an interrupt flag associated with it. The interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG) is set when the interrupt condition occurs. Each interrupt can be individually enabled by writing a '1' to the corresponding bit in the Interrupt Enable Set register (INTENSET), and disabled by writing a '1' to the corresponding bit in the Interrupt Enable Clear register (INTENCLR).

An interrupt request is generated when the interrupt flag is set and the corresponding interrupt is enabled. The interrupt request remains active until the interrupt flag is cleared, the interrupt


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

is disabled or the PDEC is reset. See the INTFLAG register description for details on how to clear interrupt flags.

The user must read the INTFLAG register to determine which interrupt condition is present.
**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

### 53.6.5. Events

The PDEC can generate the following output events:

- Overflow/Underflow: OVF
- Channel x Compare Match: MCx
- Error: ERR
- Velocity: VLC. This interrupt is available only in QDEC and HALL operation modes.
- Direction: DIR. This interrupt is available only in QDEC and HALL operation modes.

Writing a '1' to an Event Output bit in the Event Control register (EVCTRL.MCEO) enables the corresponding output event. Writing a '0' to this bit disables the corresponding output event.

**Related Links**
Event System (EVSYS)

### 53.6.6. Sleep Mode Operation

The PDEC can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in the Control A register (CTRLA.RUNSTDBY) must be written to '1'. The PDEC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

### 53.6.7. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers need to be synchronized when written or read.

The following bits are synchronized when written:

- Software Reset bit in the Control A register (CTRLA.SWRST)
- Enable bit in the Control A register (CTRLA.ENABLE)

The following registers need synchronization when written:

- Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- Status register (STATUS)
- Prescaler and Prescaler Buffer registers (PRESC and PRESCBUF)
- Compare Value x and Compare Value x Buffer registers (CCx and CCBUFx)
- Filter Value and Filter Buffer Value registers (FILTER and FILTERBUF)
- Counter Value register (COUNT)

Required write synchronization is denoted by the "Write-Synchronized" property in the register description.

The following registers are synchronized when read:

- Counter Value register (COUNT): the synchronization is done on demand through READSYNC software command (CTRLBSET.CMD)

Required read synchronization is denoted by the "Read-Synchronized" property in the register description.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Related Links**
Register Synchronization


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 53.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | RUNSTDBY | | | MODE[1:0] | | ENABLE | SWRST |
| | | 15:8 | PEREN | SWAP | | | ALOCK | CONF[2:0] | | |
| | | 23:16 | | PINVEN2 | PINVEN1 | PINVEN0 | | PINEN2 | PINEN1 | PINEN0 |
| | | 31:24 | MAXCMP[3:0] | | | | | ANGULAR[2:0] | | |
| 0x04 | CTRLBCLR | 7:0 | CMD[2:0] | | | | | | LUPD | |
| 0x05 | CTRLBSET | 7:0 | CMD[2:0] | | | | | | LUPD | |
| 0x06 | EVCTRL | 7:0 | EVEI2 | EVEI1 | EVEI0 | EVINV2 | EVINV1 | EVINV0 | EVACT[1:0] | |
| | | 15:8 | | | MCEO1 | MCEO0 | VLCEO | DIREO | ERREO | OVFEO |
| 0x08 | INTENCLR | 7:0 | | | MC1 | MC0 | VLC | DIR | ERR | OVF |
| 0x09 | INTENSET | 7:0 | | | MC1 | MC0 | VLC | DIR | ERR | OVF |
| 0x0A | INTFLAG | 7:0 | | | MC1 | MC0 | VLC | DIR | ERR | OVF |
| 0x0B | Reserved | | | | | | | | | |
| 0x0C | STATUS | 7:0 | DIR | STOP | HERR | WINERR | | MPERR | IDXERR | QERR |
| | | 15:8 | | | CCBUFV1 | CCBUFV0 | | | FILTERBUFV | PRESCBUFV |
| 0x0E | Reserved | | | | | | | | | |
| 0x0F | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x10 | SYNCBUSY | 7:0 | CC0 | COUNT | FILTER | PRESC | STATUS | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | | | | | | CC1 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | PRESC | 7:0 | | | | | PRESC[3:0] | | | |
| 0x15 | FILTER | 7:0 | FILTER[7:0] | | | | | | | |
| 0x16 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | PRESCBUF | 7:0 | | | | | PRESCBUF[3:0] | | | |
| 0x19 | FILTERBUF | 7:0 | FILTERBUF[7:0] | | | | | | | |
| 0x1A ... 0x1B | Reserved | | | | | | | | | |
| 0x1C | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | CC0 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | CC1 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | CCBUF0 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | CCBUF1 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |

## 53.8. Register Description

Registers can be 8, 16, or 32 bits wide. Atomic 8-, 16- and 32-bit accesses are supported. In addition, the 8-bit quarters and 16-bit halves of a 32-bit register, and the 8-bit halves of a 16-bit register can be accessed directly.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Some registers are optionally write-protected by the Peripheral Access Controller (PAC). Optional PAC write protection is denoted by the "PAC Write-Protection" property in each individual register description. For details, refer to Register Access Protection.

Some registers are synchronized when read and/or written. Synchronization is denoted by the "Write-Synchronized" or the "Read-Synchronized" property in each individual register description. For details, refer to Synchronization.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

### 53.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | MAXCMP[3:0] | | | | | ANGULAR[2:0] | | |
| Access | RW | RW | RW | RW | | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | PINVEN2 | PINVEN1 | PINVEN0 | | PINEN2 | PINEN1 | PINEN0 |
| Access | | RW | RW | RW | | RW | RW | RW |
| Reset | | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PEREN | SWAP | | | ALOCK | CONF[2:0] | | |
| Access | RW | RW | | | RW | RW | RW | RW |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | RUNSTDBY | | | MODE[1:0] | | ENABLE | SWRST |
| Access | | RW | | | RW | RW | RW | W |
| Reset | | 0 | | | 0 | 0 | 0 | 0 |

**Bits 31:28 – MAXCMP[3:0]** Maximum Consecutive Missing Pulses
These bits define the threshold for the maximum consecutive missing pulses in AUTOC configuration of the QDEC mode.
Outside of AUTOC configuration of QDEC mode, these bits have no effect.
These bits are not synchronized.

**Bits 26:24 – ANGULAR[2:0]** Angular Counter Length
In QDEC mode, these bits define the size of the Angular counter within COUNT. Angular counter size is equal to CTRLA.ANGULAR+9. The remaining MSB of the COUNTER register are used for counting revolutions.
For example, CTRLA.ANGULAR=0 defines the 9 LSB of COUNT as Angular counter and the residual 7 MSB of COUNT as Revolution counter. CTRLA.ANGULAR=7 will define a 16-bit Angular counter and no Revolution counter.
Outside of QDEC mode, these bits have no effect.
These bits are not synchronized.

**Table 53-1.** Angular and Revolution Counters in COUNTER Register

| ANGULAR[2:0] | Angular counter | Revolution counter |
|---|---|---|
| 0x0 | COUNTER[0:8] | COUNTER[9:15] |
| 0x1 | COUNTER[0:9] | COUNTER[10:15] |
| 0x2 | COUNTER[0:10] | COUNTER[11:15] |
| 0x3 | COUNTER[0:11] | COUNTER[12:15] |
| 0x4 | COUNTER[0:12] | COUNTER[13:15] |
| 0x5 | COUNTER[0:13] | COUNTER[14:15] |
| 0x6 | COUNTER[0:14] | COUNTER[15] |
| 0x7 | COUNTER[0:15] | no revolution counter |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 20, 21, 22 – PINVEN** IO Pin x Invert Enable
When this bit is written to '1', the corresponding input pin active level is inverted. This bit has no effect if PINENx bit is zero.
In COUNTER mode only PINVEN[0] is significant.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Pin active level is not inverted. |
| 1 | Pin active level is inverted. |

**Bits 16, 17, 18 – PINEN** PDEC Input From Pin x Enable
This bit enables the IO pin x as signal input.
In COUNTER mode, only PINVEN[0] is significant.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Event line is the signal input. |
| 1 | I/O pin is the signal input. |

**Bit 15 – PEREN** Period Enable
This bit is used to enable the CC0 register as counter period.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Period register function is disabled. |
| 1 | CC0 is acting as counter period register. |

**Bit 14 – SWAP** PDEC Phase A and B Swap
This bit is used to swap input source of signal 0 and 1.
In COUNTER mode this bit has no effect.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The input sources of signal 0 and 1 are not swapped. |
| 1 | The input sources of signal 0 and 1 are swapped. |

**Bit 11 – ALOCK** Auto Lock
When this bit is set, the Lock Update bit in Control B register (CTRLB.LUPD) is set by hardware when an UPDATE condition is detected.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Auto Lock is disabled. |
| 1 | Auto Lock is enabled. |

**Bits 10:8 – CONF[2:0]** PDEC Configuration
These bits define the PDEC configuration.
Outside of QDEC mode, these bits have no effect.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0 | X4 | Quadrature decoder direction |
| 1 | X4S | Secure Quadrature decoder direction |
| 2 | X2 | Decoder direction |
| 3 | X2S | Secure decoder direction |
| 4 | AUTOC | Auto correction mode |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 6 – RUNSTDBY** Run in Standby

This bit is used to keep the PDEC running in standby mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | The PDEC is halted in standby. |
| 1 | The PDEC continues to run in standby. |

**Bits 3:2 – MODE[1:0]** Operation Mode

These bits select one of the QDEC, HALL, COUNTER modes.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | QDEC | QDEC operating mode |
| 0x1 | HALL | HALL operating mode |
| 0x2 | COUNTER | COUNTER operating mode |

**Bit 1 – ENABLE** Enable

Due to synchronization, there is delay between writing CTRLA.ENABLE until the peripheral is enabled/disabled. The value written to CTRLA.ENABLE will read back immediately, and the Enable Synchronization Busy bit in the Synchronization Busy register (SYNCBUSY.ENABLE) will be set. SYNCBUSY.ENABLE will be cleared when the operation is complete.

| Value | Description |
|---|---|
| 0 | The peripheral is disabled. |
| 1 | The peripheral is enabled. |

**Bit 0 – SWRST** Software Reset

Writing a '0' to this bit has no effect.
Writing a '1' to this bit resets all registers in the PDEC (except DBGCTRL) to their initial state, and the PDEC will be disabled.
Writing a '1' to CTRLA.SWRST will always take precedence; all other writes in the same write-operation will be discarded.
Due to synchronization, there is a delay from writing CTRLA.SWRST until the Reset is complete. CTRLA.SWRST and SYNCBUSY.SWRST will both be cleared when the Reset is complete.

| Value | Description |
|---|---|
| 0 | There is no Reset operation ongoing. |
| 1 | A Reset operation is ongoing. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.2. Control B Clear

**Name:** CTRLBCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Set (CTRLBSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | | LUPD | |
| Access | RW | RW | RW | | | | RW | |
| Reset | 0 | 0 | 0 | | | | 0 | |

**Bits 7:5 – CMD[2:0]** Command
These bits can be used for software control of the PDEC. When a command has been executed, the CMD bit group will read back zero. The commands are executed on the next prescaled GCLK_PDEC clock cycle.
Writing a zero to this bit group has no effect.
Writing a valid value to these bits will clear the corresponding pending command.
Writing a '0' to these bits has no effect.
Writing a '1' to an individual bit will clear the corresponding bit.

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No action |
| 1 | RETRIGGER | Force a counter restart or re-trigger |
| 2 | UPDATE | Force update of double buffered registers |
| 3 | READSYNC | Force a read synchronization of COUNT |
| 4 | START | Start QDEC/HALL |
| 5 | STOP | Stop QDEC/HALL |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the PDEC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '0' to this bit has no effect.
Writing a '1' to this will disable the lock update.

| Value | Description |
|---|---|
| 0 | The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are not copied into CCx and PER registers on hardware update condition. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.3. Control B Set

**Name:** CTRLBSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Control B Clear (CTRLBCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | | | | LUPD | |
| Access | RW | RW | RW | | | | RW | |
| Reset | 0 | 0 | 0 | | | | 0 | |

**Bits 7:5 – CMD[2:0]** Command
These bits can be used for software control of the PDEC. When a command has been executed, the CMD bit group will read back zero. The commands are executed on the next prescaled GCLK_PDEC clock cycle.
Writing a zero to this bit group has no effect.
Writing a valid value to these bits will set the associated command.

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No action |
| 1 | RETRIGGER | Force a counter restart or retrigger |
| 2 | UPDATE | Force update of double buffered registers |
| 3 | READSYNC | Force a read synchronization of COUNT |
| 4 | START | Start QDEC/HALL |
| 5 | STOP | Stop QDEC/HALL |

**Bit 1 – LUPD** Lock Update
This bit controls the update operation of the PDEC buffered registers.
When CTRLB.LUPD is set, no any update of the registers with value of its buffered register is performed on hardware UPDATE condition. Locking the update ensures that all buffer registers are valid before an hardware update is performed. After all the buffer registers are loaded correctly, the buffered registers can be unlocked.
Writing a '1' to this will enable the Lock Update.

| Value | Description |
|---|---|
| 0 | The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are copied into CCx and PER registers on hardware update condition. |
| 1 | The PRESCBUF, FILTERBUF and CCBUFx buffer registers value are not copied into CCx and PER registers on hardware update condition. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.4. Event Control

**Name:** EVCTRL
**Offset:** 0x06
**Reset:** 0x0000
**Property:** Enable-Protected, PAC Write-Protection

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | MCEO1 | MCEO0 | VLCEO | DIREO | ERREO | OVFEO |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | EVEI2 | EVEI1 | EVEI0 | EVINV2 | EVINV1 | EVINV0 | EVACT[1:0] | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 12, 13 – MCEOx** Match Channel x Event Output Enable
These bits control whether event match on channel x is enabled or not and generated for every match.

| Value | Description |
|---|---|
| 0 | Match event on channel x is disabled and will not be generated. |
| 1 | Match event on channel x is enabled and will be generated for every compare. |

**Bit 11 – VLCEO** Velocity Output Event Enable
This bit is used to enable the velocity event. When enabled, an event level will be generated for each change on the qualified PDEC phases.
This bit has no effect when COUNTER operation mode is selected.

| Value | Description |
|---|---|
| 0 | VLC output event is disabled and will not be generated. |
| 1 | VLC output is enabled and will be generated for every valid velocity condition. |

**Bit 10 – DIREO** Direction Output Event Enable
This bit is used to enable the Direction event. When enabled, an event level output is generated to report the rotation direction.

| Value | Description |
|---|---|
| 0 | DIR output event is disabled and will not be generated. |
| 1 | DIR output is enabled and changes the level when the rotation direction changes. |

**Bit 9 – ERREO** Error Output Event Enable
This bit enables the output of the Error event (ERR).

| Value | Description |
|---|---|
| 0 | ERR Event output is disabled. |
| 1 | ERR Event output is enabled. |

**Bit 8 – OVFEO** Overflow/Underflow Output Event Enable
This bit is used to enable the Overflow/Underflow event. When enabled, an event will be generated when the Counter overflows/underflows.

| Value | Description |
|---|---|
| 0 | Overflow/Underflow event is disabled and will not be generated. |
| 1 | Overflow/Underflow event is enabled and will be generated for every counter overflow/underflow. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 5, 6, 7 – EVEIx** Event x Input Enable
This bit is used to enable asynchronous input event x to the counter.

| Value | Description |
|---|---|
| 0 | Incoming event x is disabled. |
| 1 | Incoming event x is enabled. |

**Bits 2, 3, 4 – EVINVx** Inverted Event x Input Enable
This bit inverts the asynchronous input event x to the counter.

| Value | Description |
|---|---|
| 0 | Input event source x is not inverted. |
| 1 | Input event source x is inverted. |

**Bits 1:0 – EVACT[1:0]** Event Action
These bits have an effect only when COUNTER operation mode is selected, and ignored in all other operation modes.
These bits define the event action the counter will perform on an event.

| Value | Name | Description |
|---|---|---|
| 0 | OFF | Event action disabled |
| 1 | RETRIGGER | Start, restart or retrigger on event |
| 2 | COUNT | Count on event |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set (INTENSET) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | VLC | DIR | ERR | OVF |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 4, 5 – MC** Channel x Compare Match Disable
Writing a '0' to MCx has no effect.
Writing a '1' to MCx will clear the corresponding Match Channel x Interrupt Disable/Enable bit, which disables the Match Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match Channel x interrupt is disabled. |
| 1 | The Match Channel x interrupt is enabled. |

**Bit 3 – VLC** Velocity Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Velocity Interrupt Disable/Enable bit, which disables the Velocity interrupt.
This bit has no effect when COUNTER operation mode is selected.

| Value | Description |
|---|---|
| 0 | The Velocity interrupt is disabled. |
| 1 | The Velocity interrupt is enabled. |

**Bit 2 – DIR** Direction Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Direction Change Interrupt Disable/Enable bit, which disables the Direction Change interrupt.
This bit has no effect when COUNTER operation mode is selected.

| Value | Description |
|---|---|
| 0 | The Direction Change interrupt is disabled. |
| 1 | The Direction Change interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Disable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Error Interrupt Disable/Enable bit, which disables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow/Underflow Interrupt Disable
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit will clear the Overflow Interrupt Disable/Enable bit, which disables the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to change this register without doing a read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Clear (INTENCLR) register.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | VLC | DIR | ERR | OVF |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 4, 5 – MC** Channel x Compare Match Enable
Writing a '0' to MCx has no effect.
Writing a '1' to MCx will set the corresponding Match Channel x Interrupt Disable/Enable bit, which enables the Match Channel x interrupt.

| Value | Description |
|---|---|
| 0 | The Match Channel x interrupt is disabled. |
| 1 | The Match Channel x interrupt is enabled. |

**Bit 3 – VLC** Velocity Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Velocity Interrupt Disable/Enable bit, which enables the Velocity interrupt.
This bit has no effect when COUNTER operation mode is selected.

| Value | Description |
|---|---|
| 0 | The Velocity interrupt is disabled. |
| 1 | The Velocity interrupt is enabled. |

**Bit 2 – DIR** Direction Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Direction Change Interrupt Disable/Enable bit, which enables the Direction Change interrupt.
This bit has no effect when COUNTER operation mode is selected.

| Value | Description |
|---|---|
| 0 | The Direction Change interrupt is disabled. |
| 1 | The Direction Change interrupt is enabled. |

**Bit 1 – ERR** Error Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Error Interrupt Disable/Enable bit, which enables the Error interrupt.

| Value | Description |
|---|---|
| 0 | The Error interrupt is disabled. |
| 1 | The Error interrupt is enabled. |

**Bit 0 – OVF** Overflow/Underflow Interrupt Enable
Writing a '0' to this bit has no effect.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing a '1' to this bit will set the Overflow Interrupt Disable/Enable bit, which enable the Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | MC1 | MC0 | VLC | DIR | ERR | OVF |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 4, 5 – MC** Channel x Compare Match
This flag is set on the next CLK_PDEC_CNT cycle after a match with the compare condition, and will generate an interrupt request if the corresponding Match Channel x Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.MCx) is '1'.
Writing a '0' to one of these bits has no effect.
Writing a '1' to one of these bits will clear the corresponding Match Channel x interrupt flag.

**Bit 3 – VLC** Velocity
This flag is set if a velocity transition occurs, and will generate an interrupt request if the Velocity Interrupt Enable bit in Interrupt Enable Set register (INTENSET.VLC) is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Velocity transition interrupt flag.
This flag is never set when COUNTER operation mode is selected.

**Bit 2 – DIR** Direction Change
This flag is set if a direction change occurs, and will generate an interrupt request if the Direction Change Interrupt Enable bit in Interrupt Enable Set register (INTENSET.DIR) is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Velocity transition interrupt flag.
This flag is never set when COUNTER operation mode is selected.

**Bit 1 – ERR** Error
This flag is set when an error condition is detected, and will generate an interrupt request if the Error Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.ERR) is '1'. The error source can be identified by reading the Status (STATUS) register.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Error interrupt flag.

**Bit 0 – OVF** Overflow/Underflow
This flag is set on the next CLK_TC_CNT cycle after an overflow condition occurs, and will generate an interrupt request if the Overflow Interrupt Enable bit in the Interrupt Enable Set register (INTENSET.OVF) is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Overflow interrupt flag.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.8. Status

**Name:** STATUS
**Offset:** 0x0C
**Reset:** 0x0040
**Property:** Read-Synchronized, Write-Synchronized

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | CCBUFV1 | CCBUFV0 | | | FILTERBUFV | PRESCBUFV |
| Access | | | R | R | | | R | R |
| Reset | | | 0 | 0 | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | DIR | STOP | HERR | WINERR | | MPERR | IDXERR | QERR |
| Access | R | R | RW | RW | | RW | RW | RW |
| Reset | 0 | 1 | 0 | 0 | | 0 | 0 | 0 |

**Bits 12, 13 – CCBUFV** Compare Channel x Buffer Valid
The bit is set when a new value is written to the corresponding CCBUF register.
The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.

**Bit 9 – FILTERBUFV** Filter Buffer Valid
This bit is set when a new value is written to the PRESCALERBUF register.
The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.
This bit is always read '0' when COUNTER operation mode is selected.

**Bit 8 – PRESCBUFV** Prescaler Buffer Valid
This bit is set when a new value is written to the PRESC register.
The bit is cleared by writing a '1' to the corresponding location or automatically cleared on an UPDATE condition.

**Bit 7 – DIR** Direction Status Flag
This bit reflects the HALL/QDEC direction.
in COUNTER mode, this bits is always read '0'.

| Value | Description |
|---|---|
| 0 | Clockwise direction. |
| 1 | Counter-clockwise direction. |

**Bit 6 – STOP** Stop
This bit reflects the HALL/QDEC decoding status.
In COUNTER mode, this bits is always read '0'.

| Value | Description |
|---|---|
| 0 | PDEC/HALL decoding is running. |
| 1 | PDEC/HALL decoding is stopped. |

**Bit 5 – HERR** Hall Error Flag
This flag is set when an invalid HALL code is detected.
The flag is cleared by writing a '1' to this bit location.
Outside of HALL mode, this bits is always read '0'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 4 – WINERR** Window Error Flag
This flag is set when the counter is outside the window monitor.
The flag is cleared by writing a '1' to this bit location.
Outside of HALL mode, this bits is always read '0'.

**Bit 2 – MPERR** Missing Pulse Error flag
This flag is set when a missing pulse error condition is detected.
The flag is cleared by writing a '1' to this bit location.
Outside of QDEC mode, this bits is always read '0'.

**Bit 1 – IDXERR** Index Error Flag
This flag is set when an index error condition is detected.
The flag is cleared by writing a '1' to this bit location.
Outside of QDEC mode, this bits is always read '0'.

**Bit 0 – QERR** Quadrature Error Flag
This flag is set when an invalid QDEC transition is detected.
The flag is cleared by writing a '1' to this bit location.
Outside of QDEC mode, this bits is always read '0'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.9. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0F
**Reset:** 0x00
**Property:** PAC Write-Protection

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Run Mode
This bit is not affected by software reset and should not be changed by software while the PDEC module is enabled.

| Value | Description |
|---|---|
| 0 | The PDEC module is halted when the device is halted in debug mode. |
| 1 | The PDEC module continues normal operation when the device is halted in debug mode. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 53.8.10. Synchronization Status

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** Read-Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CC1 |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC0 | COUNT | FILTER | PRESC | STATUS | CTRLB | ENABLE | SWRST |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7, 8 – CC** Compare Channel x Synchronization Busy
This bit is cleared when the synchronization of Compare Channel x (CCx) register between the clock domains is complete.
This bit is set when the synchronization of Compare Channel x (CCx) register between clock domains is started.

**Bit 6 – COUNT** Count Synchronization Busy
This bit is cleared when the synchronization of Count register between the clock domains is complete.
This bit is set when the synchronization of Count register between clock domains is started.

**Bit 5 – FILTER** Filter Synchronization Busy
This bit is cleared when the synchronization of Filter register between the clock domains is complete.
This bit is set when the synchronization of Filter register between clock domains is started.
This bit is always read '0' when COUNTER operation mode is selected.

**Bit 4 – PRESC** Prescaler Synchronization Busy
This bit is cleared when the synchronization of Prescaler register between the clock domains is complete.
This bit is set when the synchronization of Prescaler register between clock domains is started.

**Bit 3 – STATUS** Status Synchronization Busy
This bit is cleared when the synchronization of Status register between the clock domains is complete.
This bit is set when the synchronization of Status register between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 2 – CTRLB** Control B Synchronization Busy
This bit is cleared when the synchronization of Control B register between the clock domains is complete.
This bit is set when the synchronization of Control B register between clock domains is started.

**Bit 1 – ENABLE** Enable Synchronization Busy
This bit is cleared when the synchronization of Enable register bit between the clock domains is complete.
This bit is set when the synchronization of Enable register bit between clock domains is started.

**Bit 0 – SWRST** Software Reset Synchronization Busy
This bit is cleared when the synchronization of Software Reset register bit between the clock domains is complete.
This bit is set when the synchronization of Software Reset register bit between clock domains is started.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.11. Prescaler Value

**Name:** PRESC
**Offset:** 0x14
**Reset:** 0x00
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PRESC[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 3:0 – PRESC[3:0]** Prescaler Value
These bits select the GCLK prescaler factor.

| Value | Name | Description |
|---|---|---|
| 0 | DIV1 | No division |
| 1 | DIV2 | Divide by 2 |
| 2 | DIV4 | Divide by 4 |
| 3 | DIV8 | Divide by 8 |
| 4 | DIV16 | Divide by 16 |
| 5 | DIV32 | Divide by 32 |
| 6 | DIV64 | Divide by 64 |
| 7 | DIV128 | Divide by 128 |
| 8 | DIV256 | Divide by 256 |
| 9 | DIV512 | Divide by 512 |
| 10 | DIV1024 | Divide by 1024 |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 53.8.12. Filter Value

**Name:** FILTER
**Offset:** 0x15
**Reset:** 0x00
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FILTER[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – FILTER[7:0]** Filter Value
These bits select the PDEC inputs filter length.
These bits have no effect when COUNTER operation mode is selected.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 53.8.13. Prescaler Buffer Value

**Name:** PRESCBUF
**Offset:** 0x18
**Reset:** 0x00
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PRESCBUF[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 3:0 – PRESCBUF[3:0]** Prescaler Buffer Value
These bits hold the value of the prescaler buffer register. The value is copied in the corresponding PRESC register on UPDATE condition.

| Value | Name | Description |
|---|---|---|
| 0 | DIV1 | No division |
| 1 | DIV2 | Divide by 2 |
| 2 | DIV4 | Divide by 4 |
| 3 | DIV8 | Divide by 8 |
| 4 | DIV16 | Divide by 16 |
| 5 | DIV32 | Divide by 32 |
| 6 | DIV64 | Divide by 64 |
| 7 | DIV128 | Divide by 128 |
| 8 | DIV256 | Divide by 256 |
| 9 | DIV512 | Divide by 512 |
| 10 | DIV1024 | Divide by 1024 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.14. Filter Buffer Value

**Name:** FILTERBUF
**Offset:** 0x19
**Reset:** 0x00
**Property:** Write-Synchronized

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FILTERBUF[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – FILTERBUF[7:0]** Filter Buffer Value
These bits hold the value of the filter buffer register. The value is copied in the corresponding FILTER register on UPDATE condition.
These bits have no effect when COUNTER operation mode is selected.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 53.8.15. Counter Value

**Name:** COUNT
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Read-Synchronized, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – COUNT[15:0]** Counter Value

These bits contain the counter value. To read the most updated counter value, the READSYNC software command must be applied first (CTRLBSET.CMD = READSYNC).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.16. Channel x Compare Value

**Name:** CCx
**Offset:** 0x20 + x*0x04 [x=0..1]
**Reset:** 0x00000000
**Property:** Read-Synchronized, Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CC[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CC[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – CC[15:0]** Channel Compare Value
These bits hold value of the channel x compare register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 53.8.17. Channel x Compare Buffer Value

**Name:** CCBUFx
**Offset:** 0x30 + x*0x04 [x=0..1]
**Reset:** 0x00000000
**Property:** Write-Synchronized

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[15:8] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CCBUF[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – CCBUF[15:0]** Channel Compare Buffer Value
These bits hold the value of the channel x compare buffer register. The register is used as buffer for the associated compare register (CCx). Accessing this register using the CPU will affect the corresponding CCBVx status bit (STATUS.CCBUFVx).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 54. Electrical Characteristics at 85°C

## 54.1. Disclaimer

All typical values are measured at T = 25°C unless otherwise specified. All minimum and maximum values are valid across operating temperature and voltage unless otherwise specified.

## 54.2. Absolute Maximum Ratings

Stresses beyond those listed in the table below may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Table 54-1.** Absolute Maximum Ratings (1)

| Symbol | Description | Min. | Max. | Units |
|---|---|---|---|---|
| $V_{DD}$ | Power supply voltage | 0 | 3.8 | V |
| $I_{VDD}$ | Current into a $V_{DD}$ pin(2,3,4) | - | 60 | mA |
| $I_{GND}$ | Current out of a GND pin | - | 60 | mA |
| $V_{PIN}$ | Pin voltage with respect to GND and $V_{DD}$ | GND-0.6V | $V_{DD}$+0.6V | V |
| $T_{storage}$ | Storage temperature | -60 | 150 | °C |
| $T_{BIAS}$ | Ambient temperature under bias (up to) | -40 | 125 | °C |
| $T_J$ | Junction temperature | -40 | 145 | °C |
| **ESD qualification:** | | | | |
| $V_{HBM}$ | Human Body Model (HBM) per JESD22-A114 | - | 2000 | V |
| $V_{CDM}$ | Charged Device Model (CDM) AEC Q100-011 (ANSI/ESD STM 5.3.1)..(PKG: Mid, corner) | - | 750, 750 | V |

**Notes:**

1. Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions, above those indicated in the operation listings of this specification, is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.
2. Maximum allowable current is a function of device maximum power dissipation (See Thermal Operating Conditions).
3. For 100-pin packages: IVDD (pin 92) = 360 mA and IVDD (pin 77) = 210 mA.
4. For 128-pin packages: IVDD (pin 118) = 360 mA and IVDD (pin 97) = 210 mA.

**CAUTION** This device is sensitive to electrostatic discharges (ESD). Improper handling may lead to permanent performance degradation or malfunctioning.
Handle the device following best practice ESD protection rules: Be aware that the human body can accumulate charges large enough to impair functionality or destroy the device.

**Note:** In this data sheet $V_{DD}$ is represented as common voltage applied to both $V_{DDIO}$ and $V_{DDANA}$ pins. For additional information on terminology, refer to the section “Acronyms and Abbreviations”.

## 54.3. General Operating Ratings

The device must operate within the ratings listed below in order for all other electrical characteristics and typical characteristics of the device to be valid.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-2.** General Operating Conditions [4]

| Symbol | Description | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| $V_{DDIO}$[2,3] | IO Supply Voltage | 1.71[1] | 3.3 | 3.63 | V |
| $V_{DDIOB}$[2,3] | IOB Supply Voltage | 1.71[1] | 3.3 | 3.63 | V |
| $V_{DDANA}$[2,3] | Analog supply voltage | 1.71[1] | 3.3 | 3.63 | V |
| $V_{BAT}$ | Battery Supply Voltage | 1.71 [1] | 3.3 | 3.63 | V |

**Notes:**

1. With BOD33 is disabled, BOD12 is enabled.
2. The same voltage must be applied to $V_{DDIO}$ and $V_{DDANA}$. $V_{DDIOB}$ must be lower or equal to $V_{DDIO}$ and $V_{DDANA}$.
3. When I/O pads in the $V_{DDIOB}$ cluster are multiplexed as analog pads, $V_{DDANA}$ is used to power the I/O. Using this configuration may result in an electrical conflict if the $V_{DDIOB}$ voltage is different from that of $V_{DDIO}$ and $V_{DDANA}$. If the application has such requirements, it is required to power $V_{DDIOB}$, $V_{DDIO}$ and $V_{DDANA}$ from the same supply source to ensure that they are always at the same voltage.
4. The device must operate with in the ratings listed in the Table 58-2. Thermal Operation Conditions.

## 54.4. Injection Current

Stresses beyond those listed in the table below may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Table 54-3.** Injection Current[1, 2]

| Symbol | Description | min | Typ. | max | Unit | Comments |
|---|---|---|---|---|---|---|
| $I_{ICL}$ | Input Low Injection Current | -15 | - | - | mA | Note: 1, 3, 4<br>This Parameter applies to all pins. |
| $I_{ICH}$ | Input High Injection Current | - | - | 15 | mA | Note: 2, 3, 4<br>This parameter applies to all I/O pins. |
| $\sum I_{ICT}$ | Total Input Injection Current (Sum of all I/O and control pins) Absolute value of $\lvert\sum I_{ICT}\rvert$ | - | - | 45 | mA | Absolute instantaneous sum of all ± input injection currents from all I/O pins.( $\lvert I_{ICL}\rvert + \lvert I_{ICH}\rvert) \leq \sum I_{ICT}$ |

**Notes:**

1. VIL source < (VSS - 0.6). Characterized but not tested.
2. VIH source > (VDDIO + 0.6) for all I/O pins.
3. Injection currents > | 0 | can affect the ADC results by approximately 4 to 6 counts (i.e., VIH Source > ($V_{DDIO}$ + 0.6) or VIL source < (GND - 0.6)).
4. Any number and/or combination of I/O pins not excluded under $I_{ICL}$ or $I_{ICH}$ conditions are permitted provided the "absolute instantaneous" sum of the input injection currents from all pins do not exceed the specified $\sum I_{ICT}$ limit. To limit the injection current the user must insert a resistor in series RS between input source voltage and device pin. The resistor value is calculated according to:
   - For negative Input voltages less than (GND-0.6): RS ≥ (((GND - 0.6) - VIL source) / $I_{ICL}$)
   - For positive input voltages greater than ($V_{DDIO}$+0.6): RS ≥ ((VIH source - (VDDIO+0.6))/ $I_{ICH}$)
   - For Vpin voltages > $V_{DDIO}$ and < GND then RS = the larger of the values calculated above


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 54.5. Supply Characteristics

**Table 54-4.** Supply Rates [1]

| Symbol | Description | Fall Rate | Rise Rate | | Units |
|---|---|---|---|---|---|
| | | Max. | Min. | Max. | |
| VDDIO | Peripheral, Internal regulator and IO Supply Voltages | 50 | 0.2 | 100 | mV/µs |
| VDDIOB | | | | | |
| VDDANA | Analog Supply Voltage | | | | |

**Note:** 1. These values are based on simulation. They are not covered by production test limits or characterization.

## 54.6. Maximum Clock Frequencies

**Table 54-5.** Maximum GCLK Generator Output Frequencies (see Notes 1, 2)

| Symbol | Description | Conditions | Fmax | Units |
|---|---|---|---|---|
| $f_{GCLKGEN0}$ / $f_{GCLK_MAIN}$ (see **Note 2**) | GCLK Generator Output Frequency | undivided | 200 | MHz |
| $F_{gclkgenX}$, x={1;7} | | | 200 | MHz |
| $F_{gclkgenX}$, , x={8;11} | | | 100 | MHz |

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. GCLK Generator 0 output frequency must not exceed the AHB clock frequency. The output must be divided in case of the GCLK Generator 0 input frequency is higher than the AHB clock frequency.

**Table 54-6.** Maximum Peripheral Clock Frequencies[1]

| Symbol | Description | Max. | Units |
|---|---|---|---|
| $f_{CPU}$ | CPU clock frequency | 120 | MHz |
| $f_{AHB}$ | AHB clock frequency | 120 | MHz |
| $f_{APBx,\ x = \{A, B, C, D\}}$ | APBA, APBB, APBC and APBD clock frequency | 120 | MHz |
| $f_{GCLK_DPLLx,\ x = \{0,1\}}$ | FDPLL0 and FDPLL1 Reference clock frequency | 3.2 | MHz |
| $f_{GCLK_DPLLx_32K,\ x = \{0,1\}}$ | FDPLL0 and FDPLL1 32k Reference clock frequency | 100 | kHz |
| $f_{GCLK_DFLL48M_REF}$ | DFLL48M Reference clock frequency | 33 | kHz |
| $f_{GCLK_EIC}$ | EIC input clock frequency | 100 | MHz |
| $f_{GCLK_FREQM_MSR}$ | FREQM Measure | 200 | MHz |
| $f_{GCLK_FREQM_REF}$ | FREQM Reference | 100 | MHz |
| $f_{GCLK_EVSYS_CHANNEL_x,\ x = \{0,..,\ 11\}}$ | EVSYS channel x input clock frequency | 100 | MHz |
| $f_{GCLK_SERCOMx_SLOW,\ x = \{0, \ldots, 7\}}$ | Common SERCOMx slow input clock frequency | 12 | MHz |
| $f_{GCLK_SERCOMx_CORE,\ x = \{0, \ldots, 7\}}$ | SERCOMx input clock frequency | 100 | MHz |
| $f_{GCLK_CANx,\ x = \{0, 1\}}$ | CANx input clock frequency | 100 | MHz |
| $f_{GCLK_USB}$ | USB input clock frequency | 60 | MHz |
| $f_{GCLK_I2S}$ | I2S input clock frequency | 100 | MHz |
| $f_{GCLK_SDHCx_SLOW,\ x = \{0, 1\}}$ | Common SDHCx slow input clock frequency | 12 | MHz |
| $f_{GCLK_SDHCx_CORE,\ x = \{0, 1\}}$ | SDHCx input clock frequency | 150 | MHz |
| $f_{GCLK_TCCx,\ x = \{0, \ldots, 4\}}$ | TCCx input clock frequency | 200 | MHz |
| $f_{GCLK_TCx,\ x = \{0, \ldots, 3\}}$ | TC0, TC1, TC2, TC3 input clock frequency | 200 | MHz |
| $f_{GCLK_TCx,\ x = \{4, \ldots, 7\}}$ | TC4, TC5, TC6, TC7 input clock frequency | 100 | MHz |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 54-6.** Maximum Peripheral Clock Frequencies[1] (continued)

| Symbol | Description | Max. | Units |
|---|---|---|---|
| $f_{GCLK_PDEC}$ | PDEC input clock frequency | 200 | MHz |
| $f_{GCLK_CCL}$ | CCL input clock frequency | 100 | MHz |
| $f_{GCLK_GCLKIN}$ | External GCLK input clock frequency | 50 | MHz |
| $f_{GCLK_CM4_TRACE}$ | CM4 Trace input clock frequency | 120 | MHz |
| $f_{GCLK_AC}$ | AC digital input clock frequency | 100 | MHz |
| $f_{GCLK_ADCx,\ x = \{0, 1\}}$ | ADCx input clock frequency | 100 | MHz |
| $f_{GCLK_DAC}$ | DAC input clock frequency | 100 | MHz |

**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

## 54.7. Power Consumption

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - CPU is running on Flash with automatic wait state
  - Low-power cache enabled.
  - BOD33 is disabled
  - I/Os are inactive input mode with input trigger disabled
- **Oscillators**
  - XOSC0 (crystal oscillator) running with external 32 MHz crystal
  - XOSC32K (32 kHz crystal oscillator) running with external 32 kHz crystal in LP mode
  - FDPLL is using XOSC32K as reference
  - DFLL is using XOSC32K as reference


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-7.** Current Consumption - Active and Idle Mode

| Mode | Conditions | Regulator | Clock | $V_{DD}$ | $T_A$ | Typ | Max. | Units |
|---|---|---|---|---|---|---|---|---|
| Active | COREMARK(1) | LDO | FDPLL 120 MHz | 1.8V | Max. at 85°C Typ at 25°C | 136 | 162 | µA/MHz |
| | | | | 3.3V | | 137 | 164 | |
| | | | DFLL 48 MHz | 1.8V | | 136 | 199 | |
| | | | | 3.3V | | 136 | 199 | |
| | | | XOSC 32 MHz | 1.8V | | 146 | 243 | |
| | | | | 3.3V | | 149 | 245 | |
| | | BUCK | FDPLL 120 MHz | 1.8V | | 103 | 127 | |
| | | | | 3.3V | | 65 | 89 | |
| | | | DFLL 48 MHz | 1.8V | | 102 | 152 | |
| | | | | 3.3V | | 63 | 115 | |
| | | | XOSC 32 MHz | 1.8V | | 110 | 205 | |
| | | | | 3.3V | | 73 | 153 | |
| Idle | N/A | LDO | FDPLL 120 MHz | 1.8V | | 21 | 46 | |
| | | | | 3.3V | | 23 | 48 | |
| | | | DFLL 48 MHz | 1.8V | | 21 | 84 | |
| | | | | 3.3V | | 21 | 84 | |
| | | | XOSC 32 MHz | 1.8V | | 25 | 115 | |
| | | | | 3.3V | | 27 | 117 | |
| | | BUCK | FDPLL 120 MHz | 1.8V | | 16 | 35 | |
| | | | | 3.3V | | 11 | 28 | |
| | | | DFLL 48 MHz | 1.8V | | 16 | 63 | |
| | | | | 3.3V | | 10 | 46 | |
| | | | XOSC 32 MHz | 1.8V | | 21 | 90 | |
| | | | | 3.3V | | 19 | 71 | |

**Notes:** System Configuration used:

- MCLK all APB clocks masked except MCLK and NVMCTRL
- MCLK.AHBMASK = 0x00C00FFF
- CMCC enabled


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 54-8.** Standby Mode Current Consumption

| Mode | Conditions | Regulator Mode | $V_{DD}$ | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| Standby | Fast wake-up disabled (PM.STDBYCFG.FASTWKUP = 0x0), no peripheral running No System RAM retained (PM.STDBYCFG.RAMCFG = 0x2). 8 KB backup RAM retained | LDO | 1.8V | Max at 85°C Typ at 25°C | 43 | 870 | µA |
| | | | 3.3V | | 43 | 869 | |
| | | BUCK | 1.8V | | 26 | 570 | |
| | | | 3.3V | | 17 | 440 | |
| | Fast wake-up enabled (PM.STDBYCFG.FASTWKUP = 0x3), no peripheral running No System RAM retained (PM.STDBYCFG.RAMCFG = 0x2). 8 KB backup RAM retained | LDO | 1.8V | | 85 | 1388 | |
| | | | 3.3V | | 85 | 1392 | |
| | | BUCK | 1.8V | | 65 | 1047 | |
| | | | 3.3V | | 47 | 738 | |
| | Fast wake-up disabled (PM.STDBYCFG.FASTWKUP = 0x0), RTC running on XOSC32K No System RAM retained (PM.STDBYCFG.RAMCFG = 0x2). 8 KB backup RAM retained | LDO | 1.8V | | 43 | 870 | |
| | | | 3.3V | | 44 | 870 | |
| | | BUCK | 1.8V | | 26 | 571 | |
| | | | 3.3V | | 18 | 443 | |
| | Fast wake-up disabled (PM.STDBYCFG.FASTWKUP = 0x0), RTC running on XOSC32K 32 KB System RAM retained (PM.STDBYCFG.RAMCFG = 0x1). 8 KB backup RAM retained | LDO | 1.8V | | 45 | 912 | |
| | | | 3.3V | | 46 | 911 | |
| | | BUCK | 1.8V | | 27 | 598 | |
| | | | 3.3V | | 19 | 462 | |
| Standby | Fast wake-up disabled (PM.STDBYCFG.FASTWKUP = 0x0), RTC running on XOSC32K Full System RAM retained (PM.STDBYCFG.RAMCFG = 0x0). 8 KB backup RAM retained | LDO | 1.8V | Max at 85°C Typ at 25°C | 53 | 1068 | µA |
| | | | 3.3V | | 53 | 1067 | |
| | | BUCK | 1.8V | | 32 | 702 | |
| | | | 3.3V | | 22 | 537 | |
| | Fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), no peripheral running Full System RAM retained. | LDO | 1.8V | | 101 | 1716 | |
| | | | 3.3 V | | 101 | 1722 | |
| | | BUCK | 1.8V | | 78 | 1298 | |
| | | | 3.3V | | 55 | 911 | |
| | Fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), RTC running on XOSC32K Full System RAM retained. | LDO | 1.8V | | 102 | 1724 | |
| | | | 3.3V | | 103 | 1732 | |
| | | BUCK | 1.8V | | 79 | 1305 | |
| | | | 3.3V | | 56 | 915 | |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 54-9.** Hibernate Mode Current Consumption

| Mode | Conditions | Regulator Mode | $V_{DD}$ | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| Hibernate | No peripheral running No System RAM retained (PM.HIBCFG.RAMCFG = 0x2) No backup RAM retained (PM.HIBCFG.BRAMCFG = 0x2) | LDO | 1.8V | Max. at 85°C Typ at 25°C | 6 | 47 | µA |
| | | | 3.3V | | 6 | 48 | |
| | | BUCK | 1.8V | | 3 | 29 | |
| | | | 3.3V | | 3 | 29 | |
| | RTC is running on XOSC32K No System RAM retained (PM.HIBCFG.RAMCFG = 0x2) No backup RAM retained (PM.HIBCFG.BRAMCFG = 0x2) | LDO | 1.8V | | 6 | 48 | |
| | | | 3.3V | | 7 | 49 | |
| | | BUCK | 1.8V | | 3 | 30 | |
| | | | 3.3V | | 3 | 30 | |
| | RTC is running on XOSC32K No System RAM retained (PM.HIBCFG.RAMCFG = 0x2) 4 KB backup RAM retained (PM.HIBCFG.BRAMCFG = 0x1) | LDO | 1.8V | | 7 | 55 | |
| | | | 3.3V | | 8 | 56 | |
| | | BUCK | 1.8V | | 3 | 35 | |
| | | | 3.3V | | 4 | 33 | |
| | RTC is running on XOSC32K No System RAM retained (PM.HIBCFG.RAMCFG = 0x2) 8 KB backup RAM retained (PM.HIBCFG.BRAMCFG = 0x0) | LDO | 1.8V | | 7 | 61 | |
| | | | 3.3V | | 8 | 63 | |
| | | BUCK | 1.8V | | 4 | 39 | |
| | | | 3.3V | | 4 | 31 | |
| | RTC is running on XOSC32K 32 KB System RAM retained (PM.HIBCFG.RAMCFG = 0x1) 8 KB backup RAM retained (PM.HIBCFG.BRAMCFG = 0x0) | LDO | 1.8V | | 9 | 100 | |
| | | | 3.3V | | 10 | 101 | |
| | | BUCK | 1.8V | | 5 | 65 | |
| | | | 3.3V | | 4 | 48 | |
| | RTC is running on XOSC32K Full System RAM retained (PM.HIBCFG.RAMCFG = 0x0) 8 KB backup RAM retained (PM.HIBCFG.BRAMCFG = 0x0) | LDO | 1.8V | | 16 | 255 | |
| | | | 3.3V | | 17 | 255 | |
| | | BUCK | 1.8V | | 9 | 166 | |
| | | | 3.3V | | 7 | 121 | |

**Table 54-10.** Backup and Off Mode Current Consumption

| Mode | Conditions | $V_{DD}$ | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| Backup | Powered by VDDIO, no RTC running VDDIO+VDDANA consumption No backup RAM retained (PM.BKUPCFG.BRAMCFG = 0x2) | 1.8V | Max. at 85°C Typ at 25°C | 2.1 | 41.7 | µA |
| | | 3.3V | | 2.5 | 42.5 | |
| | Powered by VDDIO with RTC running on XOSC32K VDDIO+VDDANA consumption No backup RAM retained (PM.BKUPCFG.BRAMCFG = 0x2) | 1.8V | | 2.7 | 42.6 | |
| | | 3.3V | | 3.3 | 43.6 | |
| | Powered by VDDIO, no RTC running VDDIO+VDDANA consumption 4 KB backup RAM retained (PM.BKUPCFG.BRAMCFG = 0x1) | 1.8V | | 2.4 | 48.4 | |
| | | 3.3V | | 2.8 | 49.1 | |
| | Powered by VDDIO, no RTC running VDDIO+VDDANA consumption 8 KB backup RAM retained (PM.BKUPCFG.BRAMCFG = 0x0) | 1.8V | | 2.7 | 55.1 | |
| | | 3.3V | | 3.1 | 55.8 | |
| | Battery backup mode powered by VBAT with RTC running on XOSC32K | 1.8V | | 2.7 | 42.6 | |
| | | 3.3V | | 3.3 | 43.6 | |
| OFF | - | 1.8V | Max at 85°C Typ at 25°C | 0.191 | 2.30 | µA |
| | | 3.3V | | 0.331 | 3.35 | |

## 54.8. Wake-Up Time

Conditions:

- $V_{DD}$ = 3.3V
- LDO Regulation mode (default mode)


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

- CPU clock = DFLL48 in open loop (default configuration)
- NVM automatic wait state and cache enabled (default configuration)

**Measurement Methods**

For IDLE and STANDBY, the exit of mode is done through asynchronous EIC wake-up. The wake-up time is measured between the toggle of the EIC pin and the set of the IO pin done by the first executed instructions in EIC interrupt handler.

For Backup and hibernate, the exit of mode is done through RTC wake-up. The wake-up time is measured between the toggle of the RTC pin (SUPC_BKOUT_RTCTGL) and the set of the IO done by the first executed instructions after reset.

For OFF mode, the exit of mode is done through Reset pin, the time is measured between the rising edge of the RESETN signal and the set of the IO done by the first executed instructions after Reset.

**Table 54-11.** Wake-Up Timing

| Sleep Mode | Conditions | Typ | Unit |
|---|---|---|---|
| IDLE | | 230 | ns |
| STANDBY | STDBYCFG.FASTWKUP = 0 | 110 | µs |
| | STDBYCFG.FASTWKUP = 1 Fast Wakeup is enabled on NVM. | 92 | µs |
| | STDBYCFG.FASTWKUP = 2 Fast Wakeup is enabled on the main voltage regulator. | 25 | µs |
| | STDBYCFG.FASTWKUP = 3 Fast Wakeup is enabled on both NVM and MAINVREG. | 5 | µs |
| Hibernate | | 320 | µs |
| BACKUP | | 350 | µs |
| OFF | | 210 | µs |

## 54.9. I/O Pin Characteristics

The pins have two different speeds controlled by the Drive Strength bit located in the Pin Configuration register PORT (PORT.PINCFG.DRVSTR).

**Note:** The PA24 and PA25 pads do not have Drive Strength capability.

**Table 54-12.** I/O Pins Common Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $V_{IL}$ | Input Low-Level Voltage | $V_{DD}$ = 1.71V-3.6V | - | - | 0.3 × $V_{DD}$ | V |
| $V_{IH}$ | Input High-Level Voltage | $V_{DD}$ = 1.71V-3.6V | 0.7 × $V_{DD}$ | - | - | |
| $V_{OL}$ | Output Low-Level Voltage | $V_{DD}$ > 1.71V, $I_{OL}$ max | - | 0.1 × $V_{DD}$ | 0.2 × $V_{DD}$ | |
| $V_{OH}$ | Output High-Level Voltage | $V_{DD}$ > 1.71V, $I_{OH}$ max | 0.8 × $V_{DD}$ | 0.9 × $V_{DD}$ | - | |
| $R_{PULL}$ | Pull-up/Pull-down Resistance | - | 20 | 40 | 60 | kΩ |
| | Pull-down resistance on pads PA24 and PA25 | - | 14 | 23 | 28 | |
| $I_{LEAK}$ | Input Leakage Current | Pull-up resistors disabled | -1 | ±0.015 | 1 | µA |

**Table 54-13.** I/O Pins Maximum Output Current(2,3)

| Symbol | Parameter | Conditions | Backup Pins in Backup Mode | Normal Mode (all I/O pins) | Normal Mode (all I/O except Backup pins) | Normal mode (Backup pins) | Units |
|---|---|---|---|---|---|---|---|
| | | | | DRVSTR = 0 | DRVSTR = 1 | DRVSTR = 1 | |
| $I_{OL}$ | Maximum Output low-level current | 1.71V<= $V_{DD}$ <3.0V | 0.005 | 0.5 | 3 | 3 | mA |
| | | 3.0V <= $V_{DD}$ <= 3.63V | 0.01 | 2 | 8 | 8 | |
| $I_{OH}$ | Maximum Output high-level current | 1.71V <= $V_{DD}$ < 3.0V | 0.005 | 0.5 | 3 | 2 | |
| | | 3.0V <= $V_{DD}$ <= 3.63V | 0.01 | 2 | 8 | 6 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-14.** I/O Pins Dynamic Characteristics (see Notes 1, 2, 3 and 4)

| Symbol | Parameter | Conditions | Backup Pins in Backup Mode | Normal Mode (all I/O pins) | Normal Mode (all I/O pins) | Units |
|---|---|---|---|---|---|---|
| | | | | DRVSTR = 0 | DRVSTR = 1 | |
| $t_{RISE}$ | Maximum Rise Time | $C_{LOAD}$ = 30 pF | 4 | 0.04 | 0.01 | µs |
| $t_{FALL}$ | Maximum Fall Time | $C_{LOAD}$ = 30 pF | 4 | 0.04 | 0.01 | |

The pins with $I^2C$ alternative mode available are compliant with $I^2C$ specification.

All $I^2C$ pins support Standard mode (Sm), Fast mode (Fm), Fast-plus mode (Fm+), and High-Speed mode (HS). The available $I^2C$ pins are listed in the section "I/O Multiplexing". When an I/O pin multiplexing value is set to an $I^2C$ function, internal pull-up/pull-down resistors are disabled.

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. The PA08, PA09, PA12, PA13, PA16, PA17, PA22, PA23, PD08, PD09 pins have faster fall-time in $I^2C$ Fast Plus mode (Fm+) and High-Speed mode (HS). The fall-time can be in 1.5 ns range in Fm+ mode and in 1 ns range in HS mode.
3. The following pins are Backup pins and have different properties than normal pins: PA00, PA01, PB00, PB01, PB02, PB03, PC00, PC01.
4. The USB pads, PA24 and PA25, are compliant to the USB standard in USB mode.

## 54.10. Analog Characteristics

### 54.10.1. Voltage Regulator Characteristics

#### 54.10.1.1. Buck Converter

**Table 54-15.** Buck Converter Electrical Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $P_{EFF}$ | Power Efficiency | $I_{OUT}$ = 100µA | - | 66 | - | % |
| | | $I_{OUT}$ = 100mA | - | 74 | - | % |

**Note:** To obtain the best power efficiency from the buck regulator components the values from following table (Table 54-16) should be used.

**Table 54-16.** External Components Requirements in Switching Mode(1)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $C_{IN}$(2) | Input regulator decoupling capacitors on $V_{DDIO}$, $V_{DDIOB}$ and $V_{DDANA}$ | | - | 10 | - | µF |
| | | Ceramic dielectric X7R | - | 100 | - | nF |
| $C_{OUT}$(3) | Output regulator decoupling capacitors on $V_{DDCORE}$ | | 3.76 | 4.7 | - | µF |
| | | Ceramic dielectric X7R | - | 100 | - | nF |
| ESR $C_{OUT}$ | External Series Resistance of $C_{OUT}$ | - | - | - | 0.5 | Ω |
| $L_{EXT}$ | External inductance | | - | 10 | - | µH |
| $R_{SERIES_LEXT}$ | ESR of $L_{EXT}$ | - | - | - | 0.36 | Ω |
| $I_{SAT_L_{EXT}}$ | Saturation current | - | 500 | - | - | mA |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. It is recommended to use ceramic X7R capacitor with low-series resistance. Refer to Power Supply Connections for a typical circuit connections.
3. It is recommended to use ceramic or solid tantalum capacitor with low ESR.

### 54.10.1.2. LDO Regulator

**Table 54-17.** Decoupling Requirements

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $C_{IN}$(1) | Input regulator decoupling capacitors on $V_{DDIO}$, $V_{DDIOB}$ and $V_{DDANA}$ | - | - | 10 | - | µF |
| | | Ceramic dielectric X7R | - | 100 | - | nF |
| $C_{OUT}$(2) | Output regulator decoupling capacitors on $V_{DDCORE}$ | - | 3.76 | 4.7 | - | µF |
| | | Ceramic dielectric X7R | - | 100 | - | nF |
| ESR $C_{OUT}$ | External Series Resistance of $C_{OUT}$ | - | - | - | 0.5 | Ω |

**Notes:**

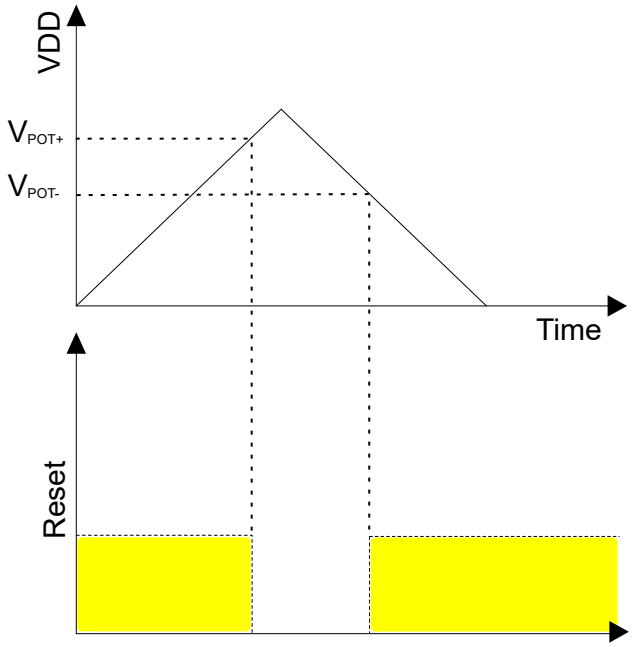
1. It is recommended to use ceramic X7R capacitor with low-series resistance. Refer to Power Supply Connections for a typical circuit connections.
2. It is recommended to use ceramic or solid tantalum capacitor with low ESR.

## 54.10.2. Power-On Reset (POR) Characteristics

**Table 54-18.** POR Characteristics

| Symbol | Parameters | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| $V_{POT+}$ | Voltage threshold Level on $V_{DDIO}$ rising | 1.53 | - | 1.64 | V |
| $V_{POT-}$ | Voltage threshold Level on $V_{DDIO}$ falling | 0.97 | - | 1.35 | V |

**Figure 54-1.** POR Operating Principle





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** The shaded area indicates that the device is in a Reset state.

### 54.10.3. Brown-Out Detectors (BOD) Characteristics

**Figure 54-2.** BOD33 Hysteresis OFF

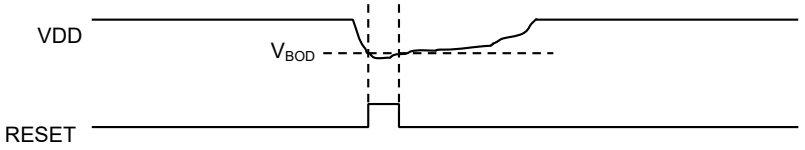



**Figure 54-3.** BOD33 Hysteresis ON

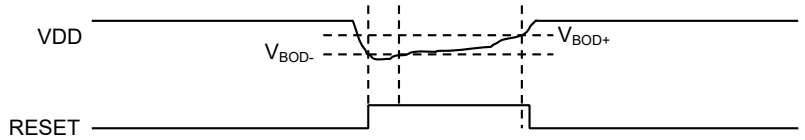



**Table 54-19.** BOD33 Characteristics on VDD and VBAT Monitoring in Normal Mode (During Power-up Phase and Active Mode)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71 V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min.(1) | Typical | Max.(2) | Units | Conditions |
| REG1 | VBOD33 | VDDANA and VBAT BOD levels | 1.463 | — | 1.555 | V | ActiveLEVEL[7:0] = 0x00 (Min Value) , HYST[3:0] =0x02 |
| | | | 1.609 | — | 1.707 | V | ActiveLEVEL[7:0] = 0x19 (Recommended Value) , HYST[3:0] =0x02 |
| | | | 1.627 | — | 1.725 | V | ActiveLEVEL[7:0] = 0x1C (Fuse Value) , HYST[3:0] = 0x02 |
| | | | 2.946 | — | 3.116 | V | ActiveLEVEL[7:0] = 0xFF(3) (Max Value) , HYST[3:0] = 0x00 |
| REG3 | $VBOD33_{LEVEL_STEP}$ | VBOD33 step size, LEVEL[7:0] | — | 6 | — | mV | |
| REG4 | $VBOD33_{HYST_STEP}$ | VBOD33 Hysteresis step size, HYST[3:0] | — | 6 | — | mV | |
| REG5 | TSTART(4) | Startup time (Time from enable to RDY) | — | 27 | — | µs | |

**Notes:**

1. BOD- = VBOD(min) = (1.5 + (LEVEL[7:0] * $VBOD33_{LEVEL_STEP}$)).
2. BOD+ = VBOD(max) = (VBOD(min) + (HYST[3:0] * $VBOD33_{HYST_STEP}$)).
3. If a hysteresis value is programmed, the following equation must be respected to avoid any overflow on VBOD33(max): LEVEL[7:0] value < 255 - HYST[3:0] value.
4. These are based on design simulation. They are not covered by production test limits or characterization.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-20.** BOD33 Characteristics on VDD and VBAT Monitoring in Low-Power Mode (During Standby/Backup/ Hibernate Modes)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71 V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min.(1) | Typical | Max.(2) | Units | Conditions |
| REG7 | VBOD33 | VDDANA and VBAT BOD levels | 1.413 | — | 1.611 | V | ActiveLEVEL[7:0] = 0x00 (Min Value) , HYST[3:0] =0x02 |
| | | | 1.551 | — | 1.773 | V | ActiveLEVEL[7:0] = 0x19 (Recommended Value) , HYST[3:0] =0x02 |
| | | | 1.569 | — | 1.791 | V | ActiveLEVEL[7:0] = 0x1C (Fuse Value) , HYST[3:0] =0x02 |
| | | | 2.845 | — | 3.23 | V | ActiveLEVEL[7:0] = 0xFF(3) (Max Value) , HYST[3:0] =0x00 |
| REG9 | $VBOD33_{LEVEL_STEP}$ | VBOD33 step size, LEVEL[7:0] | — | 6 | — | mV | |
| REG10 | $VBOD33_{HYST_STEP}$ | VBOD33 Hysteresis step size, HYST[3:0] | — | 6 | — | mV | |
| REG11 | TSTART(4) | Startup time (Time from enable to RDY) | — | 27 | — | µs | |

**Notes:**

1. BOD- = VBOD(min) = (1.5 + (LEVEL[7:0] * $VBOD33_{LEVEL_STEP}$)).
2. BOD+ = VBOD(max) = (VBOD(min) + (HYST[3:0] * $VBOD33_{HYST_STEP}$)).
3. If a hysteresis value is programmed, the following equation must be respected to avoid any overflow on VBOD33(max) : LEVEL[7:0] value < 255 - HYST[3:0] value.
4. These are based on design simulation. They are not covered by production test limits or characterization.

**Table 54-21.** BOD33 Power Consumption

| Symbol | CPU Mode | Conditions | $T_A$ | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Active / Idle | $V_{DD}$ = 1.8V | Max 85°C Typ 25°C | 8.52 | 12.07 | µA |
| | | $V_{DD}$ = 3.3V | | 10.10 | 14.28 | |
| | Standby with BOD continuous normal mode | $V_{DD}$ = 1.8V | | 4.71 | 6.34 | |
| | | $V_{DD}$ = 3.3V | | 6.01 | 8.06 | |
| | Standby with BOD continuous low power mode or Hibernate mode | $V_{DD}$= 1.8V | | 0.15 | 0.22 | |
| | | $V_{DD}$ = 3.3V | | 0.21 | 0.30 | |

## 54.10.4. Analog-to-Digital Converter (ADC) Characteristics

**Table 54-22.** ADC Electrical Specifications (1)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **Device Supply** | | | | | | | |
| ADC1 | VDDANA | ADC Module Supply | VDDANA(min) | — | VDDANA(max) | V | VDDIOx = VDDANA |
| **Reference Inputs** | | | | | | | |
| ADC3 | VREF | ADC Reference Voltage | 1 | — | VDDANA-0.4 | V | Internal or External Reference VREF ≤ VDDANA - 0.4V |
| **Analog Input Range** | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-22.** ADC Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| ADC7 | AFS | Full-Scale Analog Input Signal Range (Single-Ended) | GNDANA | — | VREF | V | |
| ADC9 | | Full-Scale Analog Input Signal Range (Differential) | -VREF | — | +VREF | V | |
| ADC11 | VCM | Input common mode voltage | 0 | | VDDANA | V | CTRLA.R2R =1 |
| | | | See Note2 | | | | CTRLA.R2R =0 |
| ADC13 | $T_{SETTING}$ | ADC stabilization Time | — | 10 | — | µs | CTRLA.ENABLE=1 or CTRLA.ONEDEMAND=1 |

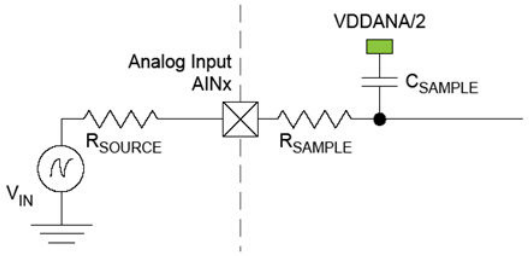
**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. Limit the input common mode voltage using the following equations (where, VCM_IN is the input channel common mode voltage):

   When CTRLA.R2R = 0:

   - VCM_IN < 0.75*VREF
   - VCM_IN > Maximum of (0, VREF-VDDANA-0.7, 1.25*VREF-VDDANA)

**Figure 54-4.** ADC Analog Input AINx




The minimum sampling time $t_{samplehold}$ for a given $R_{source}$ can be found using a general formula:

$$t_{samplehold} \geq \left(R_{sample} + R_{source}\right) \times C_{sample} \times \left(n + 2\right) \times \ln\left(2\right)$$

For 12-bit accuracy, this turns into:

$$t_{samplehold} \geq \left(R_{sample} + R_{source}\right) \times C_{sample} \times 9.7$$


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-23.** ADC Conversion Timing Requirements[4]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **ADC Clock Requirements** | | | | | | | |
| ADC21 | TAD | ADC Clock Period | 62.5 | — | 6250 | ns | |
| ADC23 | fGCLK_ADCx | ADCx Module GCLK max input freq | — | — | 100 | MHz | |
| **ADC Single-Ended Throughput Rates** | | | | | | | |
| ADC25 | FTP (Single-Ended Mode) | Throughput Rate [3] (Single-Ended) | 12.31 | — | 1231 | ksps | 12-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 13.33 | — | 1333 | | 10-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 16 | — | 1600 | | 8-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 10 | — | 1000 | ksps | 12-bit resolution, Rsource ≤6,550Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 10.67 | — | 1067 | | 10-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 12.31 | — | 1231 | | 8-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |
| **ADC Differential Mode Throughput Rates** | | | | | | | |
| ADC27 | FTP (Differential Mode) | Throughput Rate [3] (Differential Mode) | 12.31 | — | 1231 | ksps | 12-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 14.55 | — | 1455 | | 10-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 17.78 | — | 1778 | | 8-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 10 | — | 1000 | ksps | 12-bit resolution, Rsource ≤6,550Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 11.43 | — | 1143 | | 10-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 13.33 | — | 1333 | | 8-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |

**Notes:**

1. ADC_ Sample time = ((SAMPCTRL.SAMPLEN + 1) * TAD) and SAMPCTRL.OFFCOMP=0.
2. ADC_ HDW forces sample time to 4*TAD when SAMPCTRL.OFFCOMP=1, user SAMPCTRL.SAMPLEN is ignored.
3. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)). # of active analog channels used =1.
4. These values are based on simulation, and not covered by production test limits or characterization.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 54-24.** ADC Sample Timing Requirements [4]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| ADC29 | TSAMP | ADC Sample Time [1,2,3] | 1[1] | — | — | TAD | TAD(min) Ext Analog Input Rsource ≤ 147 Ω |
| | | | 2[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 2,250 Ω |
| | | | 3[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 4,400 Ω |
| | | | 4[1,2] | — | — | | TAD(min), Ext Analog Input Rsource ≤ 6,550 Ω |
| | | | 5[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 8,700 Ω |
| | | | 6[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 10,850 Ω |
| | | | See Note 5 | | | ns | Sample time with DAC as input |
| | | | 10000 | — | — | | Sample time with Temp sensor or band gap as input |
| ADC31 | TCNV | Conversion Time [3] (Single-Ended Mode) | 12 | | | TAD | 12-bit resolution |
| | | | 11 | | | | 10-bit resolution |
| | | | 9 | | | | 8-bit resolution |
| ADC33 | | Conversion Time [3] (Differential Mode) | 12 | | | | 12-bit resolution |
| | | | 10 | | | | 10-bit resolution |
| | | | 8 | | | | 8-bit resolution |
| ADC35 | CSAMPLE | ADC Internal Sample Cap | 2 | — | 3 | pf | |
| ADC37 | RSAMPLE | ADC Internal impedance | — | — | 2000 | Ω | |

**Notes:**

1. When SAMPCTRL.OFFCOMP = 0:
   - TSAMP = (((RSAMPLE + RSOURCE) * CSAMPLE * 9.7) / TAD)+1 rounded down to nearest whole integer
   - User SAMPCTRL.SAMPLEN = (TSAMP - 1)
2. When SAMPCTRL.OFFCOMP=1:
   - TSAMP = 4 (Forced by HDW)
   - User SAMPCTRL.SAMPLEN = (n/a, Ignored by HDW)
3. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)).
4. These values are based on simulation and are not covered by production test limits or characterization.
5. See ts specified in DAC Electrical Characteristics.

**Table 54-25.** Differential Mode ADC Electrical Specifications [1]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +85°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **DIFFERENTIAL MODE ADC Accuracy** | | | | | | | |
| DADC41 | Res | Resolution | 8 | — | 12 | bits | Selectable 8, 10, 12 bit Resolution Ranges |
| DADC43 | EN0B | Effective Number of bits | 10.5 | — | — | | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| DADC45 | | | 10.5 | — | — | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC47 | INL | Integral Nonlinearity | -1.5 | — | +1.5 | LSb | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| DADC49 | | | -1.5 | — | +1.5 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC51 | DNL | Differential Nonlinearity | -1.0 | — | +1.0 | | 1msps, R2R disabled,VREF = VDDANA = VDDIO = 3.0 |
| DADC53 | | | -1.0 | — | +1.0 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-25.** Differential Mode ADC Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| DADC55 | GERR | Gain Error with REFCTRL.REFCOMP=1 | - 0.18 | — | +0.16 | %FSR | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC57 | | | - 0.09 | — | +0.18 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC59 | | | - 4.3 | — | +1.8 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC61 | | | - 0.35 | — | +0.65 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC63 | | Gain Error with REFCTRL.REFCOMP=0 | - 0.2 | — | +0.16 | | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC65 | | | - 0.74 | — | +0.66 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC67 | | | - 4.9 | — | +1.6 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC69 | | | - 1 | — | +1.0 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC71 | E0FF | Offset Error with SAMPCTRL.OFFCOMP=1 | - 2.9 | — | +2.4 | mV | 1msps, Internal VREF = VDDANA = VDDIO = 3.0 |
| DADC73 | | | - 2.6 | — | +2.1 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC75 | | | - 2.3 | — | +2.3 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC77 | | | - 2.9 | — | +2.6 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC79 | | Offset Error with SAMPCTRL.OFFCOMP=0 | - 9.5 | — | +9.9 | | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC81 | | | - 9.9 | — | +9.8 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC83 | | | - 5.0 | — | +5.0 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC85 | | | - 10.8 | — | +11.3 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC87 | TUE[3] | Total Unadjusted Error | -5.2 | — | +5.2 | LSb | 1msps, R2R disabled, Internal VREF = VDDANA = VDDIO = 3.0 |
| DADC89 | | | -5.6 | — | +5.6 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| **DIFFERENTIAL MODE ADC Dynamic Performance** | | | | | | | |
| DADC91 | SINAD | Signal to Noise and Distortion | 65.3 | — | — | dB | VREF=VDDANA=VDDIO=3.0v @ 12bit max sampling rate, Fin = 14 kHz / Full range Input signal [2] |
| DADC93 | SNR | Signal to Noise ratio | 64.7 | — | — | | |
| DADC95 | SFDR | Spurious Free Dynamic Range | 76.6 | — | — | | |
| DADC97 | THD | Total Harmonic Distortion | — | — | -78.6 | | |
| DADC99 | Nrms | Noise RMS | — | — | 2.4 | mV | External VREF = 2.0V, VDDANA = VDDIO = 3.0, Constant Input Voltage |
| | | | | | 2.5 | | VREF = VDDANA = VDDIO = 3.0, Constant Input Voltage |

**Notes:**

1. These values are based on characterization. These values are not covered by test limits in production.
2. All values expressed in decibel refer to the full scale input and are tested with an input signal 0.35dB below full scale; THD measured on the first seven harmonics of the input signal.
3. With REFCTRL.REFCOMP=1 and SAMPCTRL.OFFCOMP=1.

**Table 54-26.** Single Ended Mode ADC Electrical Specifications [1]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **SINGLE ENDED MODE ADC Accuracy** | | | | | | | |
| SADC41 | Res | Resolution | 8 | — | 12 | bits | Selectable 8, 10, 12 bit Resolution Ranges |
| SADC43 | EN0B | Effective Number of bits | 8.9 | — | — | | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| SADC45 | | | 8.9 | — | — | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-26.** Single Ended Mode ADC Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| SADC47 | INL | Integral Nonlinearity | -3.2 | — | +3.2 | LSb | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| SADC49 | | | -3.0 | — | +3.0 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC51 | DNL | Differential Nonlinearity | -1.0 | — | +1.0 | | 1msps, R2R disabled,VREF = VDDANA = VDDIO = 3.0 |
| SADC53 | | | -1.0 | — | +1.1 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC55 | GERR | Gain Error with REFCTRL.REFCOMP=1 | - 0.30 | — | +0.30 | %FSR | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| SADC57 | | | - 0.20 | — | +0.25 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC59 | | | - 4.2 | — | +1.8 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| SADC61 | | | - 0.4 | — | +0.6 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| SADC71 | E0FF | Offset Error with SAMPCTRL.OFFCOMP=1 | - 18 | — | +7 | mV | 1msps, Internal VREF = VDDANA = VDDIO = 3.0 |
| SADC73 | | | - 18.6 | — | +13 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC75 | | | - 24 | — | +19 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| SADC77 | | | - 22 | — | +24 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| SADC87 | TUE[3] | Total Unadjusted Error | -18.3 | — | +18.3 | LSb | 1msps, R2R disabled, Internal VREF = VDDANA = VDDIO = 3.0 |
| SADC89 | | | -19.1 | — | +19.1 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| **SINGLE ENDED MODE ADC Dynamic Performance** | | | | | | | |
| SADC91 | SINAD | Signal to Noise and Distortion | 55.7 | — | — | dB | VREF=VDDANA=VDDIO=3.0v @ 12bit max sampling rate, Fin = 14 kHz / Full range Input signal [2] |
| SADC93 | SNR | Signal to Noise ratio | 54.7 | — | — | | |
| SADC95 | SFDR | Spurious Free Dynamic Range | 67.9 | — | — | | |
| SADC97 | THD | Total Harmonic Distortion | — | — | -65.8 | | |
| SADC99 | Nrms | Noise RMS | — | — | 2.1 | mV | External VREF = 2.0V, VDDANA = VDDIO = 3.0, Constant Input Voltage |
| | | | | | 1.7 | | VREF = VDDANA = VDDIO = 3.0, Constant Input Voltage |

**Notes:**

1. These values are based on characterization. These values are not covered by test limits in production.
2. All values expressed in decibel refer to the full scale input and are tested with an input signal 0.35dB below full scale; THD measured on the first seven harmonics of the input signal.
3. With REFCTRL.REFCOMP=1 and SAMPCTRL.OFFCOMP=1.

**Table 54-27.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ $V_{DDANA}$ | Differential mode | fs = 1 Msps / Reference buffer disabled / $V_{DDANA}$ = $V_{REF}$ = 3.0V | Max 85°C Typ 25°C | 279 | 318 | µA |
| | | fs = 1 Msps / Reference buffer enabled / $V_{DDANA}$ = 3.0V, $V_{REF}$ = 2.0V | | 482 | 653 | |
| | | fs = 10 ksps / Reference buffer disabled / $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 28 | 45 | |
| | | fs = 10 ksps / Reference buffer enabled / $V_{DDANA}$ = 3.0V, $V_{REF}$ = 2.0V | | 241 | 397 | |
| | Single Ended mode | fs = 1 Msps / Reference buffer disabled / $V_{DDANA}$ = $V_{REF}$ = 3.0V | Max 85°C Typ 25°C | 307 | 348 | µA |
| | | fs = 1 Msps / Reference buffer enabled / $V_{DDANA}$ = 3.0V, $V_{REF}$ = 2.0V | | 499 | 681 | |
| | | fs = 10 ksps / Reference buffer disabled / $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 38 | 60 | |
| | | fs = 10 ksps / Reference buffer enabled / $V_{DDANA}$ = 3.0V, $V_{REF}$ = 2.0V | | 245 | 400 | |

## 54.10.5. Digital to Analog Converter (DAC) Characteristics

**Table 54-28.** Operating Conditions [1]

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Res | Resolution | - | - | - | 12 | bits |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-28.** Operating Conditions [1] (continued)

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| clk | Internal DAC Clock frequency | - | - | - | 12 | MHz |
| fs_dac | Sampling frequency | clk/12, CCTRL=0x0 (Low Power) | - | - | 100 | ksps |
| | | clk/12, CCTRL=0x2 (High Power) | - | - | 1 | Msps |
| $V_{OUTmin}$ | Min. Output Voltage | - | - | - | 0.15 | V |
| $V_{OUTmax}$ | Max. Output Voltage | - | $V_{DDANA}$-0.15 | - | - | |
| $V_{REF}$ | External Reference input | CTRLB.REFSEL[1:0]=0x2 (VREFAB) | 1 | - | $V_{DDANA}$-0.15 | V |
| | | CTRLB.REFSEL[1:0]=0x0 (VREFAU) | 1 | - | $V_{DDANA}$ | |
| $C_{VREF}$ | External decoupling capacitor | - | - | 220 | - | nF |
| $C_{LOAD}$ | Output capacitor load | - | - | - | 50 | pF |
| $R_{LOAD}$ | Output resistance load | - | 5 | - | - | kΩ |
| $t_s$ | Settling time | For reaching ±1LSB of the final value. Step size < 500 LSB - $C_{load}$ = 50pF | - | - | 1 | µs |
| $t_{s_FS}$ | Settling time 0x080 to 0xF7F | For reaching ±1LSB of the final value. Step size from 0% to 100% - $C_{load}$ = 50pF | - | 5 | 7 | µs |

**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

**Table 54-29.** Differential Mode [1]

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| INL | Integral Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V, External Ref. = 2.0V, $C_{LOAD}$ = 50 pF | - | ±2.4 | ±3.4 | LSB |
| | | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V, Internal Ref, $C_{LOAD}$ = 50 pF | - | ±3.2 | ±4.2 | |
| DNL | Differential Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V, External Ref. = 2.0V, $C_{LOAD}$ = 50 pF | - | ±2.4 | ±3.6 | LSB |
| | | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V, Internal Ref, $C_{LOAD}$ = 50 pF | - | ±3.5 | ±5.4 | |
| Gerr | Gain Error | External Reference voltage | - | ±0.4 | ±1.7 | % FSR |
| | | 1.0V Internal Reference voltage | - | ±0.8 | ±7.0 | |
| Offerr | Offset Error | External Reference voltage | - | ±13 | ±40 | mV |
| | | 1.0V Internal Reference voltage | - | ±8 | ±64 | |
| ENOB | Effective Number Of Bits | Fs = 1 Ms/s - External Ref - CCTRL = 0x2 | 9.9 | 10.7 | 10.9 | Bits |
| SNR | Signal to Noise ratio | | 63.5 | 68.6 | 72.6 | dB |
| THD | Total Harmonic Distortion | | -79.1 | -72.5 | -61.0 | dB |

**Note:**

1. These values are based on characterization. These values are not covered by test limits in production.

**Table 54-30.** Single-Ended Mode [1]

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| INL | Integral Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V External Ref. = 2.0V, $C_{LOAD}$ = 50 pF | - | ±2.7 | ±4.0 | LSB |
| | | i12clk = 12 MHz $V_{DDANA}$ = 3.0V, Internal Ref, $C_{LOAD}$ = 50 pF | - | ±5.2 | 8.2 | |
| DNL | Differential Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V External Ref = 2.0V, $C_{LOAD}$ = 50 pF | - | ±3.5 | ±6.1 | LSB |
| | | i12clk = 12 MHz $V_{DDANA}$ = 3.0V, Internal Ref, $C_{LOAD}$ = 50 pF | - | ±6.4 | ±9.4 | |
| Gerr | Gain Error | External Reference voltage | - | ±0.3 | ±1.5 | % FSR |
| | | 1.0V Internal Reference voltage | - | ±0.8 | ±6.9 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-30.** Single-Ended Mode [1] (continued)

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Offerr | Offset Error | External Reference voltage | - | ±7 | ±21 | mV |
| | | 1.0V Internal Reference voltage | - | ±2 | ±16 | |
| ENOB | Effective Number of Bits | Fs = 1 Ms/s - External Ref - CCTRL = 0x2 | 9.1 | 10.3 | 10.7 | Bits |
| SNR | Signal to Noise Ratio | | 63.5 | 68.6 | 72.6 | dB |
| THD | Total Harmonic Distortion | | -79.1 | -72.8 | -61.0 | dB |

**Note:**

1. These values are based on characterization. These values are not covered by test limits in production.

**Table 54-31.** Power Consumption

| Symbol | Parameters | Conditions | TA | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $I_{DDANA}$ | Differential Mode, DC supply current, 2 output channels - without load | fs = 1 Msps, CCTR L= 0x2, $V_{REF}$ > 2.4V, $V_{DD}$ = 3.3V | Max. 85°C Typ. 25°C | - | 384 | 540 | µA |
| | | fs = 10 ksps, CCTRL = 0x0, $V_{REF}$ < 2.4V, $V_{DD}$ = 3.3V | | - | 283 | 411 | |
| | Single-Ended Mode, DC supply current, 2 output channels - without load | fs = 1 Msps, CCTRL = 0x2, $V_{REF}$ > 2.4V, $V_{DD}$ = 3.3V | | - | 306 | 443 | µA |
| | | fs = 10 ksps, CCTRL = 0x0, $V_{REF}$ < 2.4V, $V_{DD}$ = 3.3V | | - | 230 | 332 | |

## 54.10.6. Analog Comparator (AC) Characteristics

**Table 54-32.** Analog Comparator Characteristics

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| PNIVR | Positive and Negative input range voltage | - | 0 | - | VDDANA | V |
| ICMR | Input common mode range | - | 0 | - | VDDANA-0.2 | V |
| Off [1] | Offset | High-speed<br>COMPCTRLn.SPEED = 0x3 | -18 | +/-3 | 18 | mV |
| Vhys [2] | Hysteresis | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x0 (HYST25) | 13 | 25 | 46 | mV |
| | | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x1 (HYST50) | 30 | 50 | 71 | |
| | | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x2 (HYST75) | 39 | 75 | 101 | |
| | | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x3 (HYST100) | 64 | 100 | 132 | |
| Tpd [3] | Propagation Delay<br>Vcm = Vddana/2, Vin = +-100mV overdrive from Vcm | High-speed<br>COMPCTRLn.SPEED = 0x3 | - | 24.1 | 39 | ns |
| Tstart | Startup time | High-speed<br>COMPCTRLn.SPEED = 0x3 | - | 4.7 | 7.5 | µs |

**Notes:**

1. Hysteresis is disabled.
2. Hysteresis is enabled.
3. Tpd is measured from Vin transition to ACOUT (AC direct output) toggle.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-33.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $I_{DDANA}$ | Current consumption for One AC enabled, Hysteresis disabled voltage scaler disabled | COMPCTRLn.SPEED=0x3, $V_{DDANA}$=3.3V | Max.85°C Typ.25°C | 59 | 93 | µA |
| | Current consumption Voltage Scaler only | $V_{DDANA}$=3.3V | | 11 | 21 | |

## 54.10.7. Voltage References

**Table 54-34.** Reference Voltage Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| ADC/DAC Ref | ADC/DAC internal reference ADC.REFCTRL.REFSEL = INTREF<br>DAC.CTRLB.REFSEL = INTREF | nom. 1.0V, $V_{DDANA}$=3.3V, T= 25°C | 0.954 | 1.0 | 1.044 | V |
| | | nom. 1.1V, $V_{DDANA}$=3.3V, T= 25°C | 1.060 | 1.1 | 1.139 | |
| | | nom. 1.2V, $V_{DDANA}$=3.3V, T= 25°C | 1.150 | 1.2 | 1.248 | |
| | | nom. 1.25V, $V_{DDANA}$=3.3V, T= 25°C | 1.207 | 1.25 | 1.291 | |
| | | nom. 2.0V, $V_{DDANA}$=3.3V, T= 25°C | 1.893 | 2.0 | 2.102 | |
| | | nom. 2.2V, $V_{DDANA}$=3.3V, T= 25°C | 2.092 | 2.2 | 2.303 | |
| | | nom. 2.4V, $V_{DDANA}$=3.3V, T= 25°C | 2.282 | 2.4 | 2.513 | |
| | | nom. 2.5V, $V_{DDANA}$=3.3V, T= 25°C | 2.380 | 2.5 | 2.615 | |
| | Ref Temperature coefficient | drift over [-40, +25]°C | - | -0.01/+0.03 | - | %/°C |
| | | drift over [+25, +85]°C | - | -0.02/+0.02 | - | |
| | Ref Supply coefficient | drift over [1.71, 3.6]V | - | -0.2/+0.7 | - | %/V |
| AC Ref | AC Internal Bandgap Reference | nom.1.1V, $V_{DDANA}$ = 3.3V, T = 25°C | 1.074 | 1.1 | 1.125 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 54.11. PTC Characteristics

**Table 54-35.** Sensor Load Capacitance

| Symbol | Mode | PTC channel | Max Sensor Load (1) | Units |
|---|---|---|---|---|
| Cload | Self-capacitance | Y0 | 54 | pF |
| | | Y1 | | |
| | | Y2 | | |
| | | Y3 | | |
| | | Y4 | | |
| | | Y5 | | |
| | | Y6 | | |
| | | Y7 | | |
| | | Y8 | | |
| | | Y9 | | |
| | | Y10 | | |
| | | Y11 | | |
| | | Y12 | | |
| | | Y13 | | |
| | | Y14 | | |
| | | Y15 | | |
| | | Y16 | 51 | |
| | | Y17 | 54 | |
| | | Y18 | | |
| | | Y19 | | |
| | | Y20 | | |
| | | Y21 | 51 | |
| | | Y22 | 54 | |
| | | Y23 | | |
| | | Y24 | | |
| | | Y25 | | |
| | | Y26 | | |
| | | Y27 | | |
| | | Y28 | | |
| | | Y29 | | |
| | | Y30 | | |
| | | Y31 | | |
| | Mutual-capacitance | All | 31 | |

**Note:**

1. Capacitance load that the PTC circuitry can compensate for each channel.

**Table 54-36.** Analog Gain Settings (1) (2)

| Symbol | Setting | Average |
|---|---|---|
| Gain | GAIN_1 | 1 |
| | GAIN_2 | 2.0 |
| | GAIN_4 | 4.2 |
| | GAIN_8 | 9.1 |
| | GAIN_16 | 15.4 |
| | GAIN_32 | - |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Notes:**

1. Analog Gain is a parameter of the QTouch Library. Refer to the "QTouch Library Peripheral Touch Controller User Guide" for additional information.
2. The GAIN_16 and GAIN_32 settings are not recommended; otherwise, the PTC measurements might become unstable.

The values in the following Power Consumption table are measured values of power consumption under the following conditions:

**Operating Conditions:**

VDD = 3.0V

**Clocks:**

DFLL48M used as main clock source, running undivided at 48 MHz

CPU is running on flash with 2 wait states, at 48 MHz

PTC running at 4 MHz

**PTC Configuration**

Mutual-Capacitance mode

One touch channel

**System Configuration**

Standby Sleep mode enabled

RTC running on ULP32K: used to define the PTC scan rate, through the event system

RTC interrupts (wake up) the CPU to perform PTC scans

**Table 54-37.** Power Consumption (1)

| Symbol | Parameters | PTC scan rate (msec) | Oversamples | TA | Typ. | Max | Units |
|---|---|---|---|---|---|---|---|
| IDD | Current Consumption | 10 | 4 | Max 85°C Typ 25°C | 137 | 1164 | µA |
| | | | 16 | | 146 | 1179 | |
| | | 50 | 4 | | 77 | 1094 | |
| | | | 16 | | 79 | 1100 | |
| | | 100 | 4 | | 68 | 1086 | |
| | | | 16 | | 69 | 1089 | |
| | | 200 | 4 | | 64 | 1085 | |
| | | | 16 | | 65 | 1087 | |

**Note:**

1. These values are based on characterization.

## 54.12. NVM Characteristics

**Table 54-38.** NVM Flash Read Wait States for Worst Case Conditions (EFP part numbers)

| Wait States (WS) | 0 WS | 1 WS | 2 WS | 3 WS | 4 WS | 5 WS | 6 WS | Auto WS |
|---|---|---|---|---|---|---|---|---|
| Read Operations | 1 Cycle | 2 Cycles | 3 Cycles | 4 Cycles | 5 Cycles | 6 Cycles | 7 Cycles | n Cycles |
| CPU FMax (MHz)(1) | 19 | 38 | 57 | 76 | 95 | 100 | 120 | 120 |

**Note:**

1. VDD > 1.71V.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-39.** NVM Flash Read Wait States for Worst Case Conditions (non-EFP part numbers)

| Wait States (WS) | 0 WS | 1 WS | 2 WS | 3 WS | 4 WS | 5 WS | Auto WS |
|---|---|---|---|---|---|---|---|
| Read Operations | 1 Cycle | 2 Cycles | 3 Cycles | 4 Cycles | 5 Cycles | 6 Cycles | N Cycles |
| CPU FMax (MHz)(1) | 24 | 51 | 77 | 101 | 119 | 120 | 120 |
| CPU FMax (MHz)(2) | 22 | 44 | 67 | 89 | 111 | 120 | 120 |

**Notes:**

1. VDD > 2.7V.
2. 1.71V < VDD <= 2.7V.

Maximum operating frequencies are given in the table above in MHz, but are limited by the Embedded Flash access time when the processor is fetching code out of it. Theses tables provide the device maximum operating frequency defined by the field RWS of the NVMCTRL CTRLA register when automatic wait states (AUTOWS) is disabled. This field defines the number of Wait states required to access the Embedded Flash Memory.

**Table 54-40.** Flash Timing Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $t_{FPW}$ | Program Cycle Time | Write Page | | 1.5 | 3(1) | ms |
| $t_{CE}$ | | Chip Erase | | 6.4 | 25 (1) | s |
| $t_{FEB}$ | | Erase Block | | 50 | 200 (1) | ms |

**Note:**

1. These are based on simulation. They are not covered by production test limits or characterization.

**Table 54-41.** Flash Endurance and Data Retention

| Symbol | Parameter | Conditions | Min. | Typ. | Units |
|---|---|---|---|---|---|
| $Ret_{NVM10k}$ | Retention after up to 10k | At TA = 85°C | 20 | - | Years |
| $Cyc_{NVM}$ | Cycling Endurance(1) | At TA = 85°C | 10K | - | Cycles |

**Note:**

1. An endurance cycle is a write-and-erase operation.

**Table 54-42.** Flash Erase and Programming Current

| Symbol | Parameter | Typ. | Max. | Units |
|---|---|---|---|---|
| $I_{FAP}$ | Active Current current during whole programming operation | 8 | | mA |
| $I_{FAE}$ | Active Current current during Erase operation | 8 | | mA |

## 54.13. Oscillators Characteristics

### 54.13.1. Crystal Oscillator (XOSC) Characteristics

#### Digital Clock Characteristics

The following table provides the characteristics for the oscillator when a digital clock is applied on XIN.

**Table 54-43.** Digital Clock Characteristics

| Symbol | Parameter | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| $F_{XIN}$ | XIN clock frequency | - | - | 48 | MHz |
| $DC_{XIN}$ (see **Note 1**) | XIN clock duty cycle | 40 | | 60 | % |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

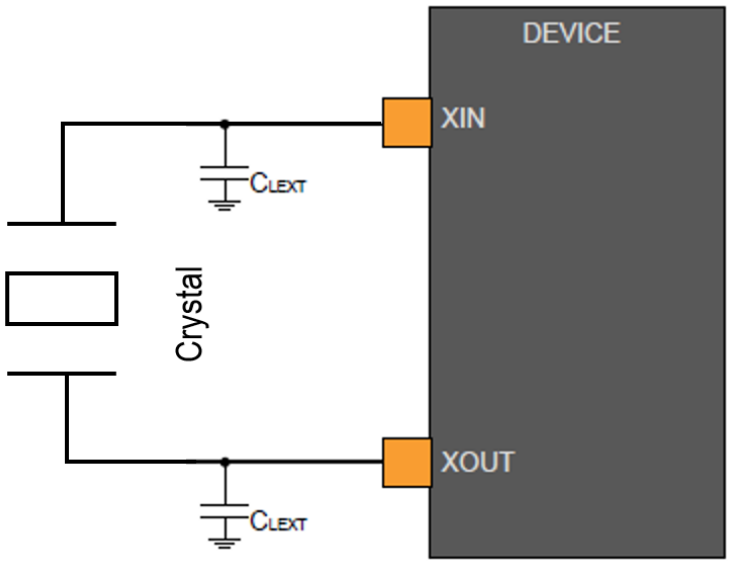
**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

### Crystal Oscillator Characteristics

The following Table provides the characteristics for the oscillator when a crystal is connected between XIN and XOUT.

**Figure 54-5.** Oscillator Connection




The user must choose a crystal oscillator where the crystal load capacitance $C_L$ is within the range given in the Table. The exact value of $C_L$ can be found in the crystal data sheet. The capacitance of the external capacitors ($C_{LEXT}$) can then be computed as follows:

### Load Capacitance Equation

$$C_{LOAD} = ([C_{XIN} + C_{LEXT}] * [C_{XOUT} + C_{LEXT}]) / ([C_{XIN} + C_{LEXT} + C_{LEXT} + C_{XOUT}]) + C_{STRAY}$$

**Where:**

$C_{LOAD}$ = Crystal Mfg. $C_{LOAD}$ specification.

$C_{XIN}$ = XOSC XIN pin data sheet specification.

$C_{XOUT}$ = XOSC XOUT pin data sheet specification.

$C_{LEXT}$ = Required external crystal load capacitor.

$C_{STRAY}$ (Osc PCB capacitance) = 1.5 pf per 12.5 mm (0.5 inches) (TRACE W = 0.175 mm, H = 36 µm, T = 113 µm).

**Table 54-44.** Multi-Crystal Oscillator Electrical Characteristics [2]

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{OUT}$ | Crystal oscillator frequency | | 8 | - | 48 | MHz |
| $C_L$ | Crystal Load | F = 8 MHz | - | - | 20 | pF |
| | | F = 16 MHz | - | - | 20 | |
| | | F = 32 MHz | - | - | 13 | |
| | | F = 48 MHz | - | - | 13 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-44.** Multi-Crystal Oscillator Electrical Characteristics [2] (continued)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| ESR | Crystal Equivalent Series Resistance - SF=3 | F = 8 MHz, $C_L$=20 pF - IMULT = 0x3 | - | - | 181 | Ω |
| | | F = 16 MHz, $C_L$ = 20 pF - IMULT = 0x4 | - | - | 180 | |
| | | F= 24 MHz, $C_L$ = 20 pF - IMULT = 0x5 | - | - | 70 | |
| | | F = 48 MHz, $C_L$ = 13 pF - IMULT = 0x6 | - | - | 70 | |
| $C_{XIN}$ | XOSC XIN pin capacitance | - | - | 6.3 | - | pF |
| $C_{XOUT}$ | XOSC XOUT pin capacitance | - | - | 5.9 | - | |
| $D_L$ | Drive Level [1] | ENALC = ON | - | - | 100 | µW |
| Tstart | Startup time | F = 8MHz - CL=20 pF - Cshunt = 2 pF - IMULT=0x3 | - | 39700 | 72200 | Cycles |
| | | F = 16MHz - CL=20 pF - Cshunt = 1,5 pF - IMULT=0x4 | - | 37550 | 62000 | |
| | | F = 24MHz - CL=20 pF - Cshunt = 2,5 pF - IMULT=0x5 | - | 32700 | 68500 | |
| | | F = 48MHz - CL=13 pF - Cshunt = 5 pF - IMULT=0x6 | - | 18400 | 38500 | |

**Notes:**

1. To ensure that the crystal is not overdriven, the automatic loop control is recommended to be turned ON (ENALC = 1).
2. Maximum start up time that can be programmed in XOSCCTRL.STARTUP is 1s.

**Table 54-45.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | F = 8 MHz - $C_L$ = 20 pF - IMULT = 0x3, ENALC = OFF | Max. 85°C, Typ. 25°C | 0.43 | 1.02 | mA |
| | | ENALC = ON | | 0.16 | 0.66 | |
| | | F = 16 MHz - $C_L$ = 20 pF - IMULT = 0x5, ENALC = OFF | | 1.31 | 2.39 | |
| | | ENALC = ON | | 0.25 | 0.81 | |
| | | F = 32 MHz - $C_L$ = 13 pF - IMULT = 0x5, ENALC = OFF | | 2.92 | 4.75 | |
| | | ENALC = ON | | 0.40 | 1.09 | |
| | | F = 48 MHz - $C_L$ = 13 pF - IMULT = 0x6, ENALC = OFF | | 2.70 | 4.79 | |
| | | ENALC = ON | | 0.76 | 1.46 | |

## 54.13.2. External 32 kHz Crystal Oscillator (XOSC32K) Characteristics

### Digital Clock Characteristics

The following table describes the characteristics for the oscillator when a digital clock is applied on XIN32 pin.

**Table 54-46.** Digital Clock Characteristics[1]

| Symbol | Parameter | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| $f_{CPXIN32}$ | XIN32 clock frequency | | 32.768 | | kHz |
| $DC_{XIN}$ | XIN32 clock duty cycle | | 50 | | % |

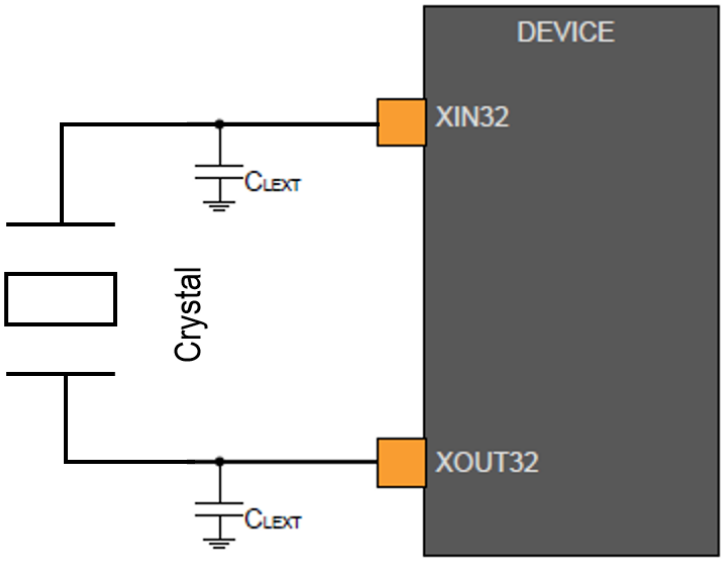
**Note:** 1.These values are based on simulation. They are not covered by production test limits or characterization.

### Crystal Oscillator Characteristics

The following section describes the characteristics for the oscillator when a crystal is connected between XIN32 and XOUT32 pins.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 54-6.** Oscillator Crystal Connection




The user must choose a crystal oscillator where the crystal load capacitance $C_L$ is within the range given in the table. The exact value of $C_L$ can be found in the crystal data sheet. The capacitance of the external capacitors ($C_{LEXT}$) can then be computed as follows:

**Load Capacitance Equation**

$C_{LOAD} = ([C_{XIN} + C_{LEXT}] * [C_{XOUT32K} + C_{LEXT}]) / ([C_{XIN32K} + C_{LEXT} + C_{LEXT} + C_{XOUT32K}]) + C_{STRAY}$

**Where:**

$C_{LOAD}$ = Crystal Mfg. $C_{LOAD}$ specification.

$C_{XIN32k}$ = XIN32 capacitance, see the following table.

$C_{XOUT32k}$ = XOUT32 pin capacitance, see the following table.

$C_{LEXT}$ = Required external crystal load capacitor.

$C_{STRAY}$ (Osc PCB capacitance) = 1.5 pf per 12.5 mm (0.5 inches) (TRACE W = 0.175 mm, H = 36 µm, T = 113 µm).

**Table 54-47.** 32 kHz Crystal Oscillator Electrical Characteristics [2]

| Symbol | Parameter | Conditions | | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $F_{OUT}$[1] | Crystal oscillator frequency | - | | - | 32.768 | - | kHz |
| CL[1] | Crystal load capacitance | - | | - | - | 12.5 | pF |
| ESR | Crystal Equivalent Series Resistance - SF=3 | f=32.768 kHz, $C_L$=12.5 pF | Std. Gain | - | - | 58 | kΩ |
| | | | High Gain | - | - | 90 | |
| $C_{XIN32k}$ | XIN32 pin capacitance | - | | - | 3.1 | - | pF |
| $C_{XOUT32k}$ | XOUT32 pin capacitance | - | | - | 3.2 | - | |

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. Maximum start up time that can be programmed in OSC32KCTRL.STARTUP is 8s.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-48.** Power Consumption

| Symbol | Parameter | Conditions | Ta | Gain Mode | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current consumption | VDD=3.0V | Max 85°C Typ 25°C | Std. | 1.5 | 2 | µA |
| | | | | High | 1.9 | 3 | |

### 54.13.3. Internal Ultra Low Power 32 kHz RC Oscillator (OSCULP32K) Characteristics

**Table 54-49.** Ultra-Low-Power Internal 32 kHz RC Oscillator Electrical Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $F_{OUT}$ | Output frequency | [-40, +85]°C, VDDANA>1.71V | 27.12 | 32.768 | 37.68 | kHz |
| Duty(1) | Duty Cycle | | - | 50 | - | % |

**Note:** These values are based on simulation. They are not covered by production test limits or characterization.

### 54.13.4. Digital Frequency Locked Loop (DFLL48M) Characteristics

**Table 54-50.** DFLL48M Characteristics - Open Loop Mode (1)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{OpenOUT}$ | Output frequency | DFLLVAL after Reset<br>LDO Regulator mode, [-40, 85]°C | 45.8 | 48 | 49.3 | MHz |
| | | DFLLVAL after Reset<br>LDO Regulator mode, [0, 60]°C | 47.2 | 48 | 48.81 | |
| $T_{OpenSTARTUP}$ | Startup time | DFLLVAL after Reset<br>$F_{OUT}$ within 90% of final value | - | 4.3 | 7 | µs |

**Note:**

1. DFLL48 in open loop can be used only with LDO regulator.

**Table 54-51.** DFLL48M Characteristics - Closed Loop Mode

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{CloseOUT}$ | Average Output frequency | $f_{REF}$ = XTAL, 32.768 kHz, 100 ppm<br>DFLLMUL = 1464 | - | 47.972 | - | MHz |
| $F_{REF}$(1,2) | Input reference frequency | - | 732 | 32768 | 33000 | Hz |
| $F_{CloseJitter}$ | Period Jitter | $f_{REF}$ = XTAL, 32.768 kHz, 100 ppm<br>DFLLMUL = 1464 | - | - | 0.42 | ns |
| $T_{Lock}$ | Lock time | $F_{REF}$ = XTAL, 32.768 kHz, 100 ppm<br>DFLLMUL = 1464<br>DFLLVAL after Reset<br>DFLLCTRL.BPLCKC = 1<br>DFLLCTRL.QLDIS = 0<br>DFLLCTRL.CCDIS = 1<br>DFLLMUL.FSTEP = 10 | - | 429 | 1145 | µs |

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. To ensure that the device stays within the maximum allowed clock frequency, any reference clock for the DFLL in close loop must be within 2% error accuracy.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-52.** DFLL48M Power Consumption

| Symbol | Parameter | Conditions | Ta | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | Open Loop mode - DFLLVAL after reset VDD = 3.3V | Max. 85°C Typ. 25°C | - | 404 | 854 | µA |
| | | Closed Loop mode - $f_{REF}$ = 32 .768 kHz VDD = 3.3V | | - | | | |

### 54.13.5. Fractional Digital Phase Lock Loop (FDPLL) Characteristics

**Table 54-53.** Fractional Digital Phase Lock Loop Characteristics (2)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $f_{IN}$(1) | Input Frequency | | 32 | - | 3200 | kHz |
| $f_{OUT}$(1) | Output Frequency | | 96 | - | 200 | MHz |
| Jp | Period jitter (Peak-Peak value) | $f_{IN}$ = 32 kHz, $f_{OUT}$ = 96 MHz | - | 1.9 | 2.7 | % |
| | | $f_{IN}$ = 32 kHz, $f_{OUT}$ = 200 MHz | - | 3.4 | 4.9 | |
| | | $f_{IN}$ = 3.2 MHz, $f_{OUT}$ = 96 MHz | - | 2.0 | 3.0 | |
| | | $f_{IN}$ = 3.2 MHz, $f_{OUT}$ = 200 MHz | - | 4.3 | 6.6 | |
| $t_{LOCK}$ | Lock Time | After startup, time to get lock signal. $f_{IN}$ = 3.2 MHz | - | 54 | 95 | µs |
| Duty (1) | Duty cycle | - | - | 50 | - | % |

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. These FDPLL200Mn characteristics are applicable with LDO regulator and a direct reference (i.e., REFCLK is XOSC or XOSC32K, not GCLK).

**Table 54-54.** Fractional Digital Phase Lock Loop Power Consumption

| Symbol | Parameter | Conditions | TA | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | Clk = 96 MHz, VDD = 3.3V | Max. 85°C Typ. 25°C | 0.9 | 1.3 | mA |
| | | Clk = 200 MHz, VDD = 3.3V | | 2 | 2.3 | |

## 54.14. Timing Characteristics

### 54.14.1. External Reset Characteristics

**Table 54-55.** External Reset Characteristics(1)

| Symbol | Parameter | Min. | Units |
|---|---|---|---|
| $t_{EXT}$ | Minimum Reset pulse width | 1 | µs |

**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

**Related Links**

Multiplexed Signals


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 54.14.2. SERCOM in SPI Mode Timing

**Figure 54-7.** SPI Timing Requirements in Host Mode

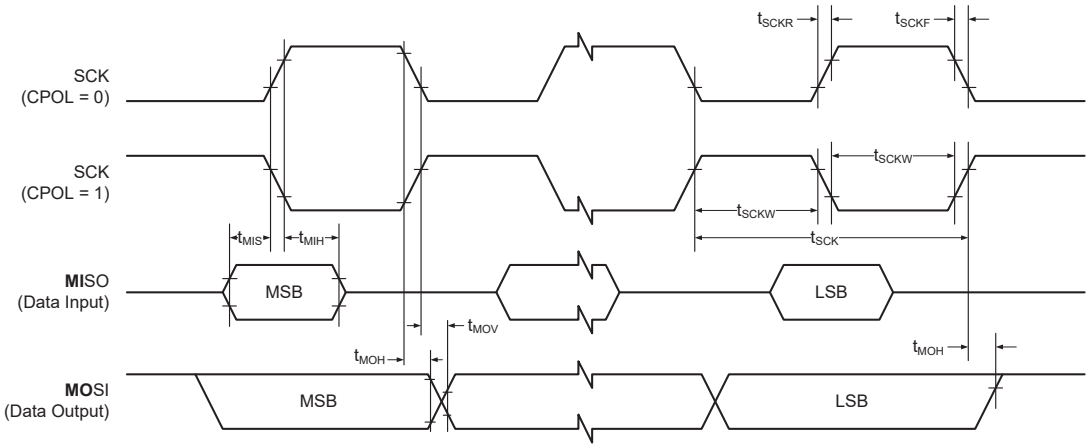
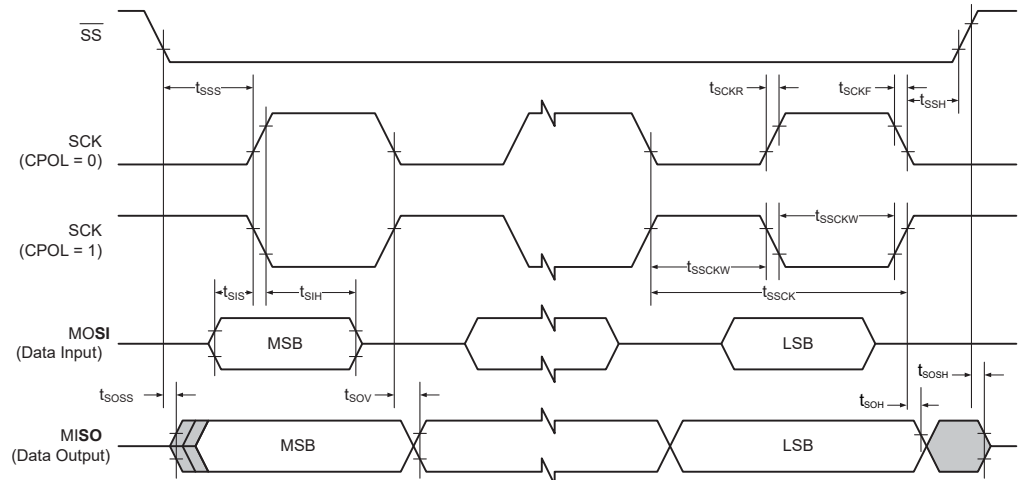



**Figure 54-8.** SPI Timing Requirements in Client Mode





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-56.** SPI Timing Characteristics and Requirements(1)

| Symbol | Parameter | Conditions | | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $t_{SCK}$(10) | SCK period | Host | Reception | 2*($t_{MIS}$+$t_{CLIENT_OUT}$)(3) | - | - | ns |
| | | Host | Transmission | 2*($t_{MOV}$+$t_{CLIENT_IN}$)(4) | - | - | |
| $t_{SCKW}$ | SCK high/low width | Host | | - | 0.5*$t_{SCK}$ | | |
| $t_{SCKR}$ | SCK rise time(2) | Host, VDD>2.70V | | - | - | 2.68 | |
| | | Host, VDD>1.71V | | | | 5.41 | |
| $t_{SCKF}$ | SCK fall time(2) | Host, VDD>2.70V | | - | - | 2.81 | |
| | | Host, VDD>1.71V | | | - | 5.84 | |
| $t_{MIS}$ | MISO setup to SCK | Host, VDD>2.70V | | 18 | - | - | |
| | | Host, VDD>1.71V | | 19 | - | - | |
| $t_{MIH}$ | MISO hold after SCK | Host,VDD>2.70V | | 0 | - | - | |
| | | Host, VDD>1.71V | | 0 | - | - | |
| $t_{MOV}$ | MOSI output valid SCK | Host, VDD>2.70V | | - | - | 9 | |
| | | Host, VDD>1.71V | | - | - | 14 | |
| $t_{MOH}$ | MOSI hold after SCK | Host, VDD>2.70V | | -3 | - | - | |
| | | Host, VDD>1.71V | | -3 | - | - | |
| $t_{SSCK}$ | Client SCK Period | Client | Reception | 2*($t_{SIS}$+$t_{HOST_OUT}$)(5) | - | - | ns |
| | | Client | Transmission | 2*($t_{SOV}$+$t_{HOST_IN}$)(6) | - | - | |
| $t_{SSCKW}$ | SCK high/low width | Client | | - | 0.5*$t_{SSCK}$ | - | |
| $t_{SSCKR}$ | SCK rise time(2) | Client | | - | 0.25*$t_{SSCK}$ | - | |
| $t_{SSCKF}$ | SCK fall time(2) | Client | | - | 0.25*$t_{SSCK}$ | - | |
| $t_{SIS}$ | MOSI setup to SCK | Client, VDD>2.70V | | 7.5 | - | - | |
| | | Client, VDD>1.71V | | 8.5 | - | - | |
| $t_{SIH}$ | MOSI hold after SCK | Client, VDD>2.70V | | 4 | - | - | |
| | | Client, VDD>1.71V | | 4 | - | - | |
| $t_{SSS}$ | $\overline{SS}$ setup to SCK | Client | PRELOADEN=1 | $t_{SOSS}$+$t_{EXT_MIS}$+2*$t_{APBC}$(8)(9) | - | - | |
| | | | PRELOADEN=0 | $t_{SOSS}$+$t_{EXT_MIS}$(8) | - | - | |
| $t_{SSH}$ | $\overline{SS}$ hold after SCK | Client | | 0.5*$t_{SSCK}$ | - | - | |
| $t_{SOV}$ | MISO output valid SCK | Client, VDD>2.70V | | 15 | - | - | |
| | | Client, VDD>1.71V | | 24 | - | - | |
| $t_{SOH}$ | MISO hold after SCK | Client, VDD>2.70V | | 0 | - | - | |
| | | Client, VDD>1.71V | | 0 | - | - | |
| $t_{SOSS}$ | MISO setup after $\overline{SS}$ low | Client, VDD>2.70V | | - | - | 1* $t_{SCK}$ | |
| | | Client, VDD>1.71V | | - | - | 1* $t_{SCK}$ | |

1. These values are based on simulation, with capacitance load between 5 pF and 20 pF. These values are not covered by test limits in production.
2. Refer to the section “I/O Pin Characteristics”.
3. Where $t_{CLIENT_OUT}$ is the Client external device output response time, generally $t_{EXT_SOV}$+$t_{LINE_DELAY}$ (7).
4. Where $t_{CLIENT_IN}$ is the Client external device input constraint, generally $t_{EXT_SIS}$+$t_{LINE_DELAY}$ (7).
5. Where $t_{CLIENT_OUT}$ is the Host external device output response time, generally $t_{EXT_MOV}$+$t_{LINE_DELAY}$ (7).
6. Where $t_{CLIENT_IN}$ is the Host external device input constraint, generally $t_{EXT_MIS}$+$t_{LINE_DELAY}$ (7).
7. $t_{LINE_DELAY}$ is the transmission line time delay.
8. $t_{EXT_MIS}$ is the input constraint for the Host external device.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

9. $t_{APBC}$ is the APB period for SERCOM.
10. When the integrity of communication is required to maintain both transmission and reception, the maximum SPI clock frequency must be the lower value of the reception or transmission mode maximum frequency as shown in the following equations.
    - Reception: $t_{SCK} = 2*(t_{MIS}+t_{CLIENT_OUT}) = 2*(18 + 8) = 52nS$
    - Transmission: $t_{SCK} = 2*(t_{MOV}+t_{CLIENT_IN}) = 2*(9 + 20) = 58nS$

### 54.14.3. QSPI Characteristics

**Figure 54-9.** QSPI SDR Host Mode 0

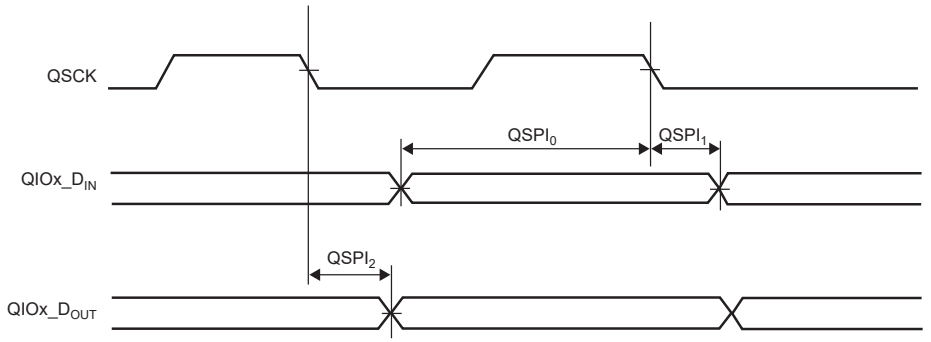



**Figure 54-10.** QSPI SDR Host Mode 1

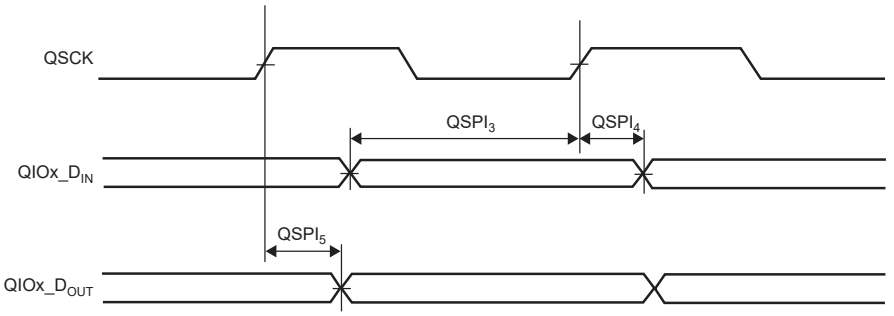
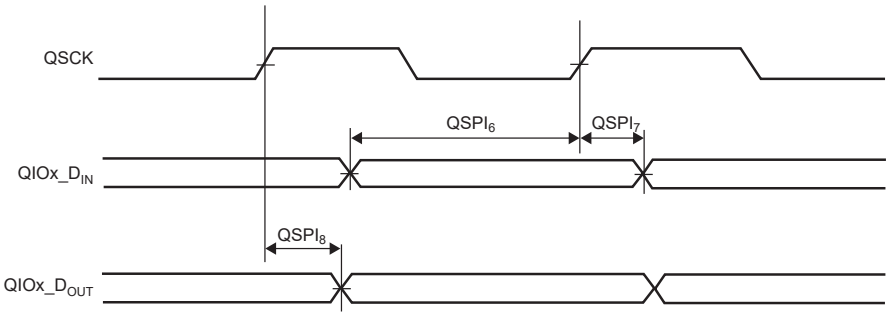



**Figure 54-11.** QSPI SDR Host Mode 2





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 54-12.** QSPI SDR Host Mode 3

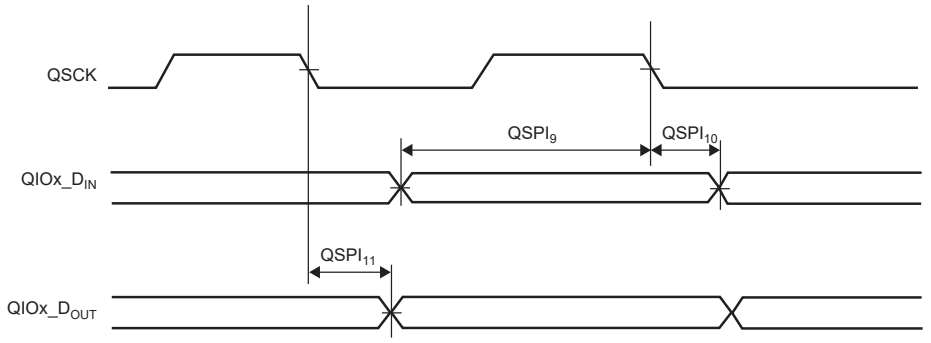



**Figure 54-13.** QSPI DDR Mode 0 READ

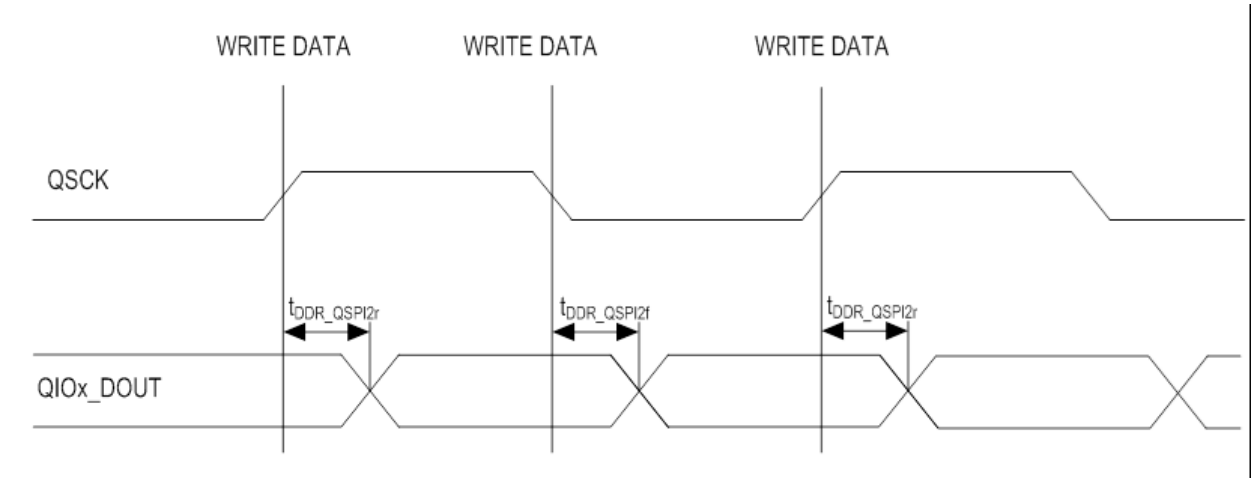



**Figure 54-14.** QSPI DDR Mode 0 WRITE

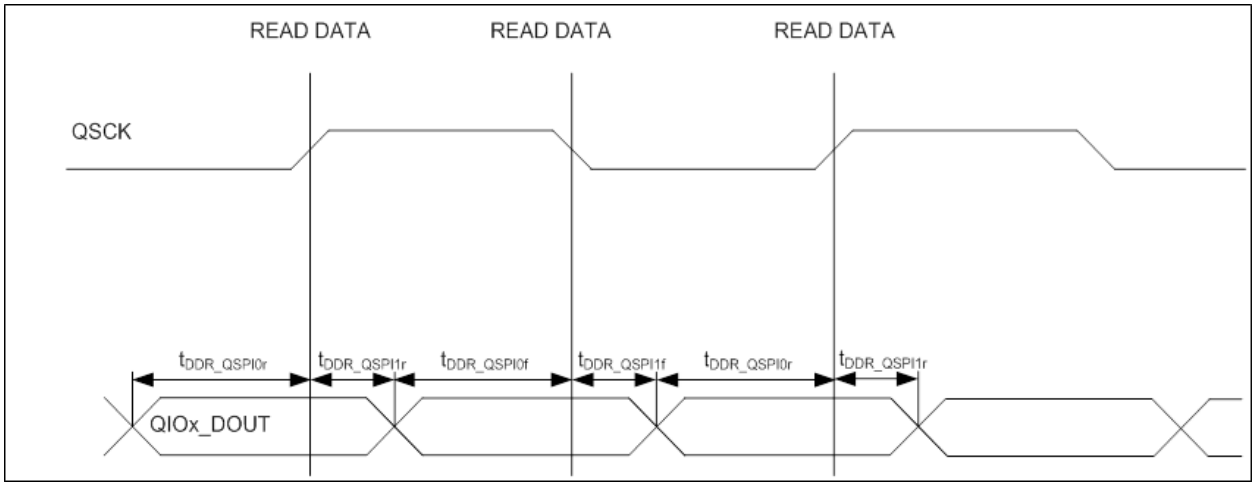



**Table 54-57.** QSPI Timing Characteristics (see Note 1)

| Name | Description | Mode | $V_{DD}$ = 1.8V | | | $V_{DD}$ = 3.3V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| $f_{SDR_m0_m2}$ | QSPI SDR Frequency | Host SDR Mode 0/2 | - | - | 50.0 | - | - | 75 | MHz |
| $f_{SDR_m1_m3}$ | QSPI SDR Frequency | Host SDR Mode 1/3 | - | - | 50.0 | - | - | 50 | |
| $f_{DDR}$ | QSPI DDR Frequency | Host mode | - | - | 37.5 | - | - | 66 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-57.** QSPI Timing Characteristics (see Note 1) (continued)

| Name | Description | Mode | $V_{DD}$ = 1.8V | | | $V_{DD}$ = 3.3V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| $t_{SDR_QSPI0}$ | Input Setup Time | Host SDR mode 0 | 3.86 | - | - | 3.85 | - | - | ns |
| $t_{SDR_QSPI1}$ | Input Hold Time | Host SDR mode 0 | 0.00 | - | - | 0.19 | - | - | |
| $t_{SDR_QSPI2}$ | Data Out Valid Time | Host SDR mode 0 | - | - | 3.33 | - | - | 2.67 | |
| $t_{SDR_QSPI3}$ | Input Setup Time | Host SDR mode 1 | 3.79 | - | - | 3.59 | - | - | ns |
| $t_{SDR_QSPI4}$ | Input Hold Time | Host SDR mode 1 | 0.06 | - | - | 0.19 | - | - | |
| $t_{SDR_QSPI5}$ | Data Out Valid Time | Host SDR mode 1 | - | - | 2.71 | - | - | 2.71 | |
| $t_{SDR_QSPI6}$ | Input Setup Time | Host SDR mode 2 | 3.79 | - | - | 3.58 | - | - | ns |
| $t_{SDR_QSPI7}$ | Input Hold Time | Host SDR mode 2 | 0.06 | - | - | 0.19 | - | - | |
| $t_{SDR_QSPI8}$ | Data Out Valid Time | Host SDR mode 2 | - | - | 2.74 | - | - | 2.65 | |
| $t_{SDR_QSPI9}$ | Input Setup Time | Host SDR mode 3 | 3.86 | - | - | 3.86 | - | - | ns |
| $t_{SDR_QSPI10}$ | Input Hold Time | Host SDR mode 3 | -0.10 | - | - | 0.19 | - | - | |
| $t_{SDR_QSPI11}$ | Data Out Valid Time | Host SDR mode 3 | - | - | 3.22 | - | - | 2.60 | |
| $t_{DDR_QSPI0f}$ | Input Setup Time | Host DDR mode 0 fall edge | 3.87 | - | - | 3.85 | - | - | ns |
| $t_{DDR_QSPI1f}$ | Input Hold Time | Host DDR mode 0 fall edge | 0.00 | - | - | 0.19 | - | - | |
| $t_{DDR_QSPI2f}$ | Data Out Valid Time | Host DDR mode 0 fall edge | - | - | 2.1 | - | - | 2.03 | |
| $t_{DDR_QSPI0r}$ | Input Setup Time | Host DDR mode 0 rise edge | 3.81 | - | - | 3.57 | - | - | |
| $t_{DDR_QSPI1r}$ | Input Hold Time | Host DDR mode 0 rise edge | 0.06 | - | - | 0.19 | - | - | |
| $t_{DDR_QSPI2r}$ | Data Out Valid Time | Host DDR mode 0 rise edge | - | - | 3.13 | - | - | 2.12 | |

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. All timing characteristics are given for 20pF capacitive load.

**Table 54-58.** QSPI Maximum Frequency examples(1)

| QSPI Mode | CLK_QSPI2X _AHB | CLK_QSPI _AHB | Max. CPU_CLK | Max. QSPI Speed | Conditions |
|---|---|---|---|---|---|
| SDR | X | 120 MHz | 120 MHz | 60 MHz(2) | BAUD -> BAUD[7:0] must be greater than 0 to ensure QSPI clock frequency is as per electrical specifications provided in table 54-57. |
| | X | 75 MHz | 75 MHz | 75 MHz | - |
| DDR | 132 MHz | 66 MHz | 66 MHz | 66 MHz | - |

**Notes:**

1. Examples shown do not supersede the electrical specifications shown in Table 54-57. QSPI Timing Characteristics.
2. CPU clock frequency(CPU_CLK) should always be multiple of SQI frequency (CLK_QSPI_AHB).

### 54.14.4. GMAC Characteristics

#### Timing Conditions

**Table 54-59.** GMAC Load Capacitance on Data, Clock Pads

| Symbol | Description | Condition | Min. | Max. | Units |
|---|---|---|---|---|---|
| $C_L$ | Load Capacitance | $V_{DD}$=3.3V | 0 | 20 | pF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### Timing Constraints

The GMAC must be constrained so as to satisfy the timings of standards given the following two tables, in MAX corner.

**Figure 54-15.** Minimum and Maximum Access Time of GMAC Output Signals

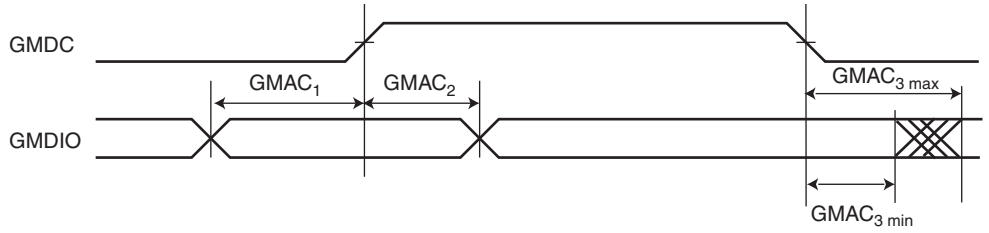



**Table 54-60.** Minimum and Maximum Access Time of GMAC Output Signals

| Symbol | Parameter | Min. | Max. | Units |
|---|---|---|---|---|
| $GMAC_1$ | Setup for GMDIO from GMDC rising | 10 | - | ns |
| $GMAC_2$ | Hold for GMDIO from GMDC rising | 10 | - | |
| $GMAC_3$ | GMDIO toggling from GMDC falling | 0(1) | 10(1) | |

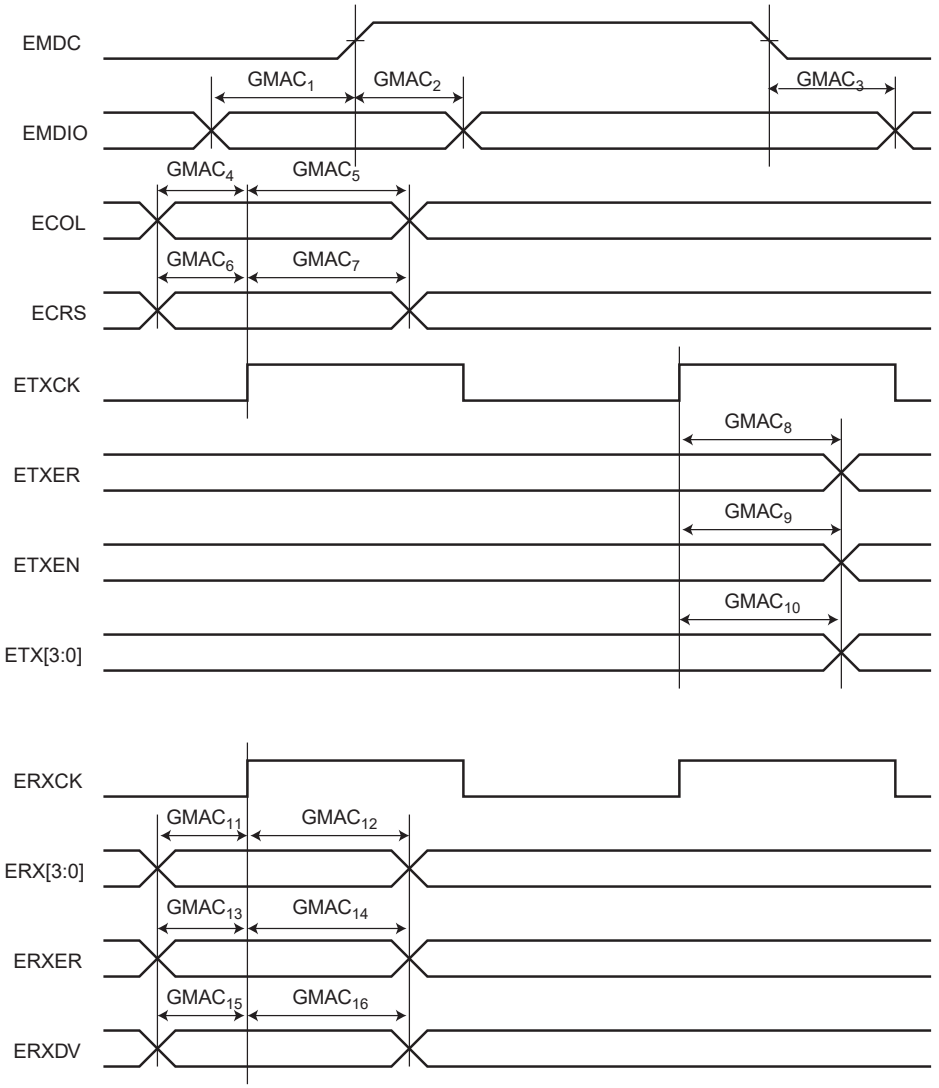
**Note:**

1. For GMAC output signals, min. and max. access times are defined:
   - The min. access time is the time between the GMDC falling edge and the signal change.
   - The max. access time is the time between the GMDC falling edge and the signal stabilizes.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## MII Mode

**Figure 54-16.** GMAC MII Mode Signals





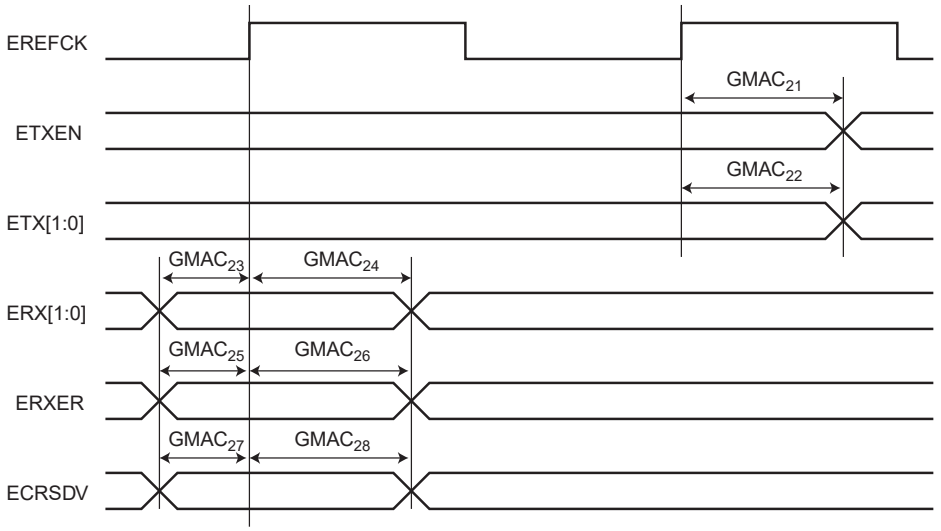
© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-61.** GMAC MII Mode Timings

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $GMAC_4$ | Setup for GCOL from GTXCK rising | 10 | – | ns |
| $GMAC_5$ | Hold for GCOL from GTXCK rising | 10 | – | |
| $GMAC_6$ | Setup for GCRS from GTXCK rising | 10 | – | |
| $GMAC_7$ | Hold for GCRS from GTXCK rising | 10 | – | |
| $GMAC_8$ | GTXER toggling from GTXCK rising | 10 | 25 | |
| $GMAC_9$ | GTXEN toggling from GTXCK rising | 10 | 25 | |
| $GMAC_{10}$ | GTX toggling from GTXCK rising | 10 | 25 | |
| $GMAC_{11}$ | Setup for GRX from GRXCK | 10 | – | |
| $GMAC_{12}$ | Hold for GRX from GRXCK | 10 | – | |
| $GMAC_{13}$ | Setup for GRXER from GRXCK | 10 | – | |
| $GMAC_{14}$ | Hold for GRXER from GRXCK | 10 | – | |
| $GMAC_{15}$ | Setup for GRXDV from GRXCK | 10 | – | |
| $GMAC_{16}$ | Hold for GRXDV from GRXCK | 10 | – | |

## RMIII Mode

**Figure 54-17.** GMAC RMII Mode Signals





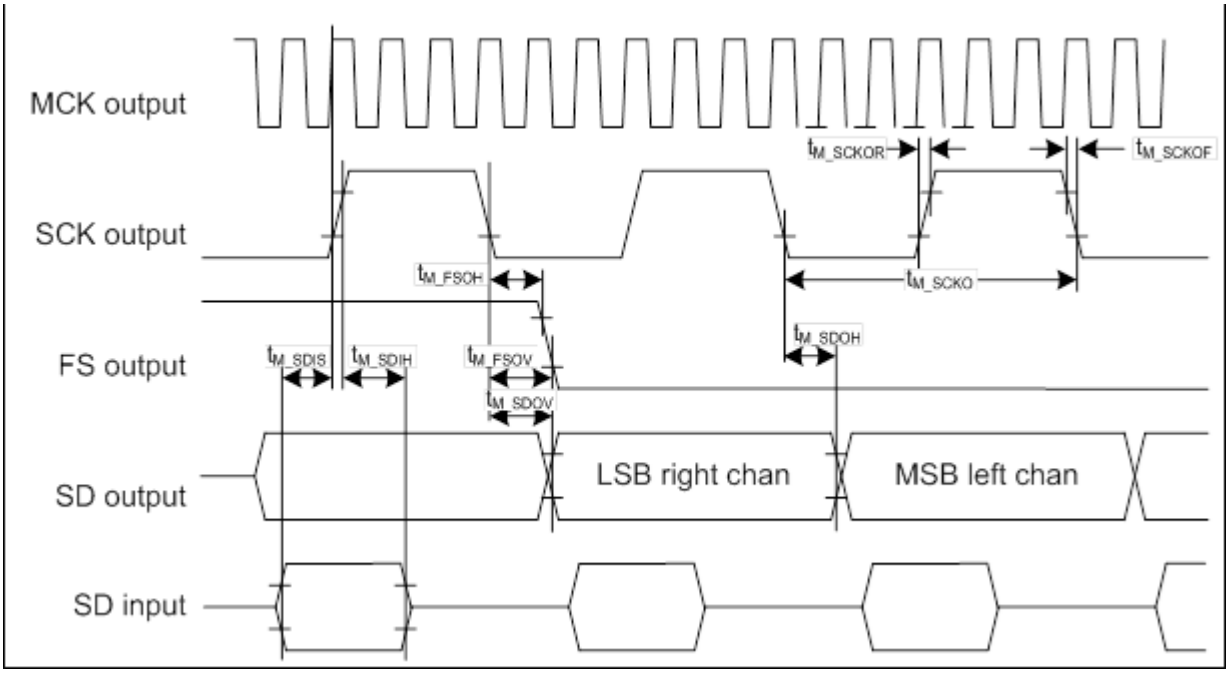
© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-62.** GMAC RMII Mode Timings

| Symbol | Parameter | Min. | Max. | Units |
|---|---|---|---|---|
| $GMAC_{21}$ | ETXEN toggling from EREFCK rising | 2 | 16 | ns |
| $GMAC_{22}$ | ETX toggling from EREFCK rising | 2 | 16 | |
| $GMAC_{23}$ | Setup for ERX from EREFCK rising | 4 | - | |
| $GMAC_{24}$ | Hold for ERX from EREFCK rising | 2 | - | |
| $GMAC_{25}$ | Setup for ERXER from EREFCK rising | 4 | - | |
| $GMAC_{26}$ | Hold for ERXER from EREFCK rising | 2 | - | |
| $GMAC_{27}$ | Setup for ECRSDV from EREFCK rising | 4 | - | |
| $GMAC_{28}$ | Hold for ECRSDV from EREFCK rising | 2 | - | |

## 54.14.5. $I^2S$ Characteristics

**Figure 54-18.** Host Mode: SCK, FX, and MCK are Output





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 54-19.** Client Mode: SCK and FS are Input

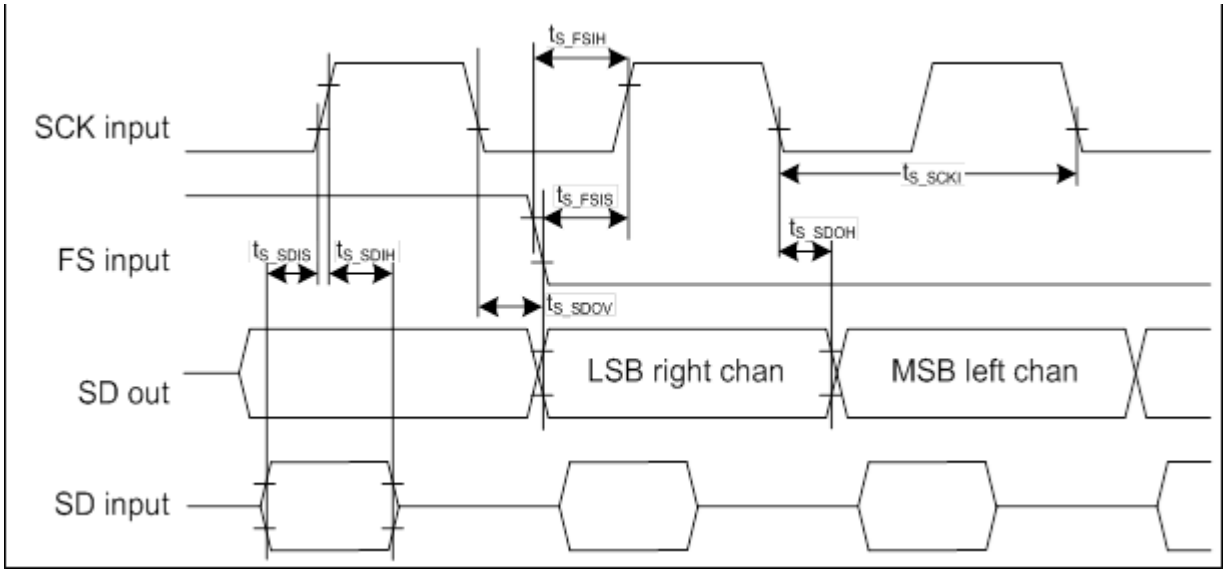



**Figure 54-20.** PDM2 Mode

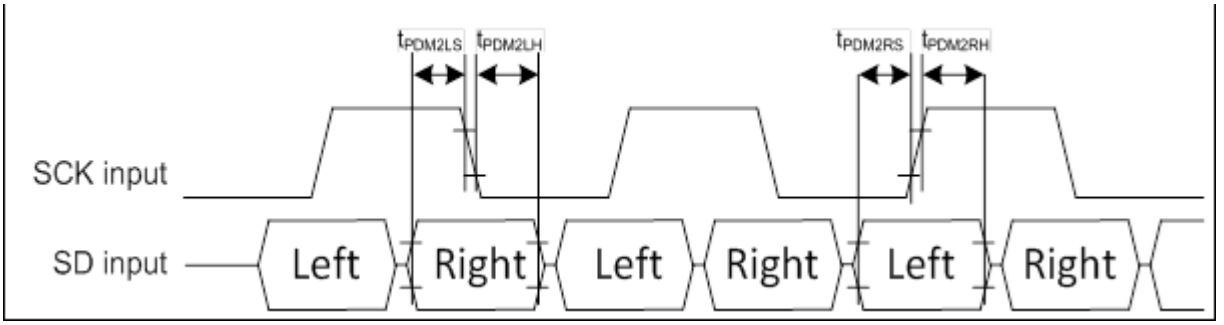



**Table 54-63.** $I^2S$ Timing Characteristics and Requirements (see Note 1)

| Name | Description | Mode | VDD = 1.8V | | | VDD = 3.3V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| $t_{M_MCKOR}$ | $I^2S$ MCK rise time(2) | Host mode / Capacitive load CL = 20 pF | - | - | 5.41 | - | - | 2.68 | ns |
| $t_{M_MCKOF}$ | $I^2S$ MCK fall time (2) | Host mode / Capacitive load CL = 20 pF | - | - | 5.84 | - | - | 2.81 | ns |
| $d_{M_MCKO}$ | $I^2S$ MCK duty cycle | Host mode | - | 50.0 | - | - | 50.0 | - | % |
| $d_{M_MCKI}$ | $I^2S$ MCK duty cycle | Host mode, pin is input (1b) | - | 50.0 | - | - | 50.0 | - | % |
| $t_{M_SCKOR}$ | $I^2S$ SCK rise time (2) | Host mode / Capacitive load CL = 20 pF | - | - | 5.06 | - | - | 2.51 | ns |
| $t_{M_SCKOF}$ | $I^2S$ SCK fall time (2) | Host mode / Capacitive load CL = 20 pF | - | - | 5.46 | - | - | 2.64 | ns |
| $d_{M_SCKO}$ | $I^2S$ SCK duty cycle | Host mode | - | 50.0 | - | - | 50.0 | - | % |
| $f_{M_SCKO}$ $1/t_{M_SCKO}$ | $I^2S$ SCK frequency | Host mode Supposing external device response delay is 30ns | - | - | 10.97 | - | - | 12.07 | MHz |
| $f_{S_SCKI}$ $1/t_{S_SCKI}$ | $I^2S$ SCK frequency | Client mode Supposing external device response delay is 30ns | - | - | 15.63 | - | - | 15.87 | MHz |
| $d_{S_SCKO}$ | $I^2S$ SCK duty cycle | Client mode | - | 50.0 | - | - | 50.0 | - | % |
| $t_{M_FSOV}$ | FS valid time | Host mode | - | - | 5.4 | - | - | 4.2 | ns |
| $t_{M_FSOH}$ | FS hold time | Host mode | -0.3 | - | - | -0.3 | - | - | ns |
| $t_{S_FSIS}$ | FS setup time | Client mode | 7.8 | - | - | 7.5 | - | - | ns |
| $t_{S_FSIH}$ | FS hold time | Client mode | 0.0 | - | - | 0.0 | - | - | ns |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-63.** $I^2S$ Timing Characteristics and Requirements (see Note 1) (continued)

| Name | Description | Mode | VDD = 1.8V | | | VDD = 3.3V | | | Units |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | Min. | Typ. | Max. | |
| $t_{M_SDIS}$ | Data input setup time | Host mode | 15.8 | - | - | 11.6 | - | - | ns |
| $t_{M_SDIH}$ | Data input hold time | Host mode | 3.4 | - | - | 3.4 | - | - | ns |
| $t_{S_SDIS}$ | Data input setup time | Client mode | 2.4 | - | - | 1.9 | - | - | ns |
| $t_{S_SDIH}$ | Data input hold time | Client mode | -1.1 | - | - | -1.0 | - | - | ns |
| $t_{M_SDOV}$ | Data output valid time | Host transmitter | - | - | 3.7 | - | - | 3.0 | ns |
| $t_{M_SDOH}$ | Data output hold time | Host transmitter | -0.5 | - | - | -0.5 | - | - | ns |
| $t_{S_SDOV}$ | Data output valid time | Client transmitter | - | - | 16.4 | - | - | 12.1 | ns |
| $t_{S_SDOH}$ | Data output hold time | Client transmitter | 4.1 | - | - | 4.1 | - | - | ns |
| $t_{PDM2LS}$ | Data input setup time | Host mode PDM2 Left | 15.8 | - | - | 11.6 | - | - | ns |
| $t_{PDM2LH}$ | Data input hold time | Host mode PDM2 Left | 3.4 | - | - | 3.4 | - | - | ns |
| $t_{PDM2RS}$ | Data input setup time | Host mode PDM2 Right | 15.1 | - | - | 11.6 | - | - | ns |
| $t_{PDM2RH}$ | Data input hold time | Host mode PDM2 Right | 3.4 | - | - | 3.4 | - | - | ns |

**NOTICE** All timing values are given for 20 pF capacitive load.

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. Refer to the section “I/O Pin Characteristics”.

### 54.14.6. PCC Characteristics

Speed requirements for all 8/10/12/14-bits are:

- pclk: 48 MHz at 3.3V
- pclk: 28 MHz at 1.8V

APB clock minimum is 2 × N pclk


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Figure 54-21.** PCC Signaling

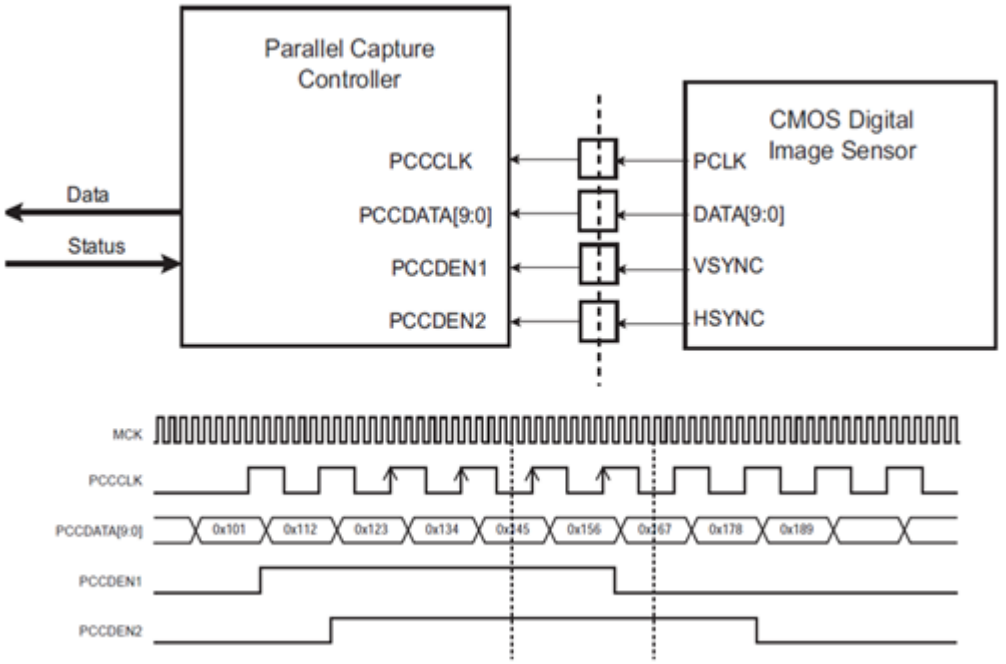
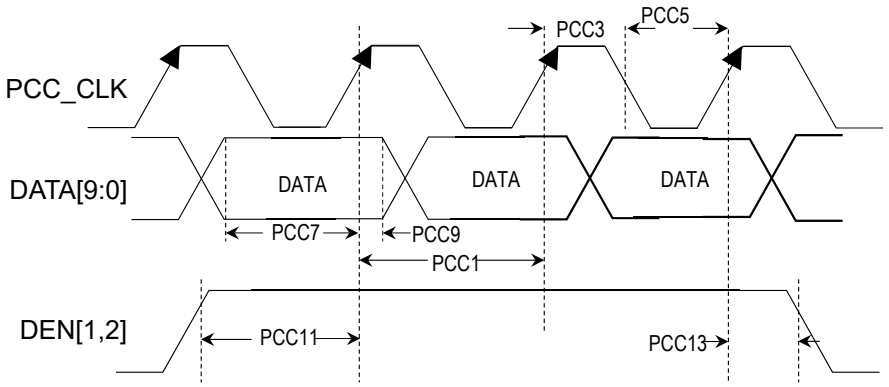



**Figure 54-22.** PCC - Peripheral Capture Controller Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-64.** PCC - Peripheral Capture Controller Electrical Specifications [4]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>-40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| PCC1 | PCC_CLK | External PCC Data Input Clock Period | 20 | — | — | ns | VDDIOx(min) |
| PCC3 | PCCHIGH | PCC Data Clock input High time | 10 | — | — | ns | |
| PCC5 | PCCLOW | PCC Data Clock input Low time | 10 | — | — | ns | |
| PCC7 | PCCDSETUP | Clock to Data Setup Time | 3 | — | — | ns | |
| PCC9 | PCCDHOLD | Clock to Data Hold Time | 2 | — | — | ns | |
| PCC11 | PCCDENBSETUP [2] | Data Enable [1,2] Setup Time [2] | 3 | — | — | ns | |
| PCC13 | PCCDENBHOLD [3] | Data Enable [1,2] Hold Time [3] | 2 | | | | |
| PCC15 | CLK_APB_PCC[1] | Internal MCLK PCC Module Clock [1] | — | — | fCPU | Mhz | |

**Notes:**

1. CLK_APB_PCC ≥ (2 * PCC_CLK).
2. PCCDENBSETUP ≥ PCC7.
3. PCCDENBHOLD ≥ PCC9.
4. These values are based on simulation. They are not covered by production test limits or characterization.

## 54.14.7. TC - Timer Counter Electrical Specifications

**Figure 54-23.** TCx Timer Capture Input Module Timing Diagrams

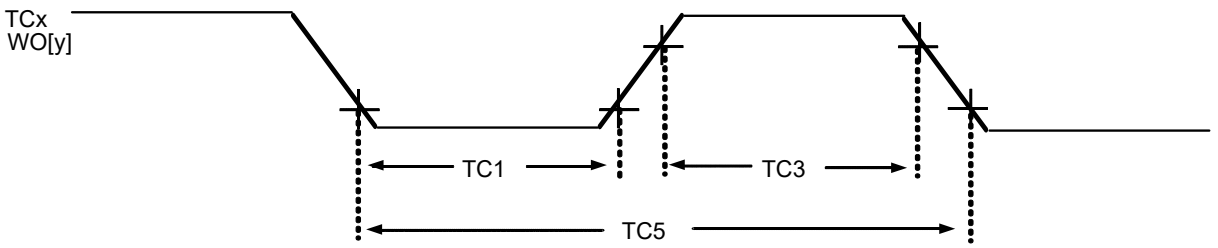
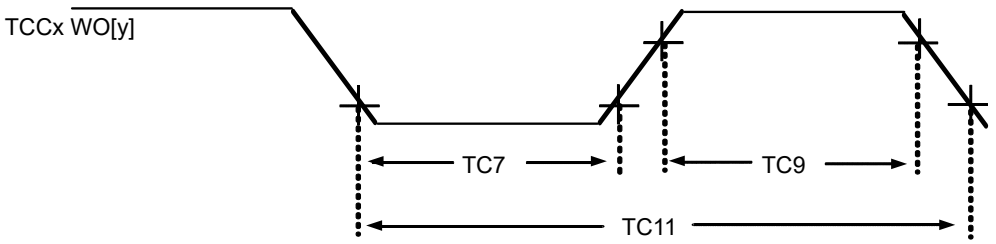



**Figure 54-24.** TCx Timer Compare Output Module Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 54-65.** TCx Timer Capture Module Electrical Specifications [1]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA = 1.71 V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics[1] | Min. | Typical | Max. | Units | Conditions |
| TC1 | TCINLOW | Capture TCx Input Low Time | 2/ fGCLK_TCx | — | — | ns | VDDIOx(min) and meet TC5 spec |
| TC3 | TCINHIGH | Capture TCx Input High Time | 2/ fGCLK_TCx | — | — | ns | VDDIOx(min) and meet TC5 spec |
| TC5 | TCINPERIOD | Capture Input Period | 4/ fGCLK_TCx | — | — | ns | VDDIOx(min) |
| TC7 | TCOUTLOW | Compare TCx Output Low Time | 1/ fGCLK_TCx | — | — | ns | VDDIOx(min) and meet TC11 spec |
| TC9 | TCOUTHIGH | Compare TCx Output High Time | 1/ fGCLK_TCx | — | — | ns | VDDIOx(min) and meet TC11 spec |
| TC11 | TCOUTPERIOD | Compare Output Period | 2/ fGCLK_TCx | — | — | ns | VDDIOx(min) |
| TC13 | fGCLK_TCx | GCLK clock for TCx | — | — | 200<br>100 | MHz | VDDIOx(min), for TC0,1,2,3<br>VDDIOx(min), for TC4,5,6,7 |

**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

## 54.14.8. TCC - Timer Counter Controller Electrical Specifications

**Figure 54-25.** TCCx Timer Capture Input Module Timing Diagrams

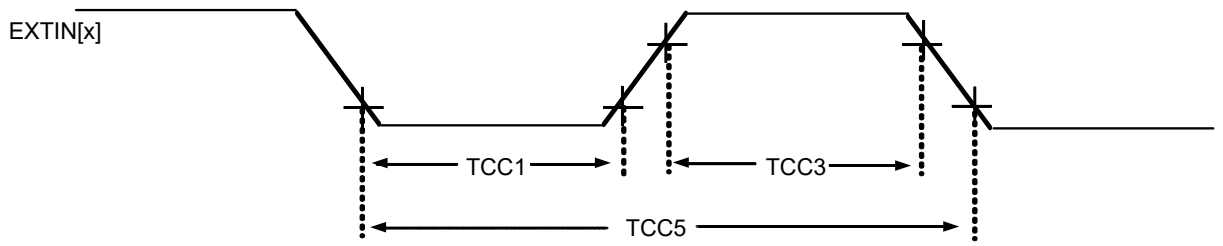
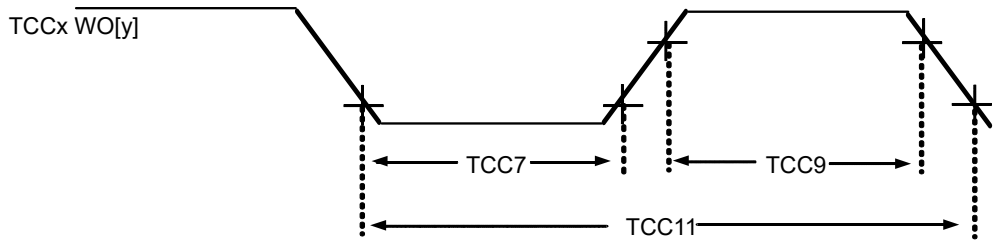



**Figure 54-26.** TCCx Timer Compare Output Module Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Figure 54-27.** TCCx Timer Compare Fault Output Module Timing Diagrams

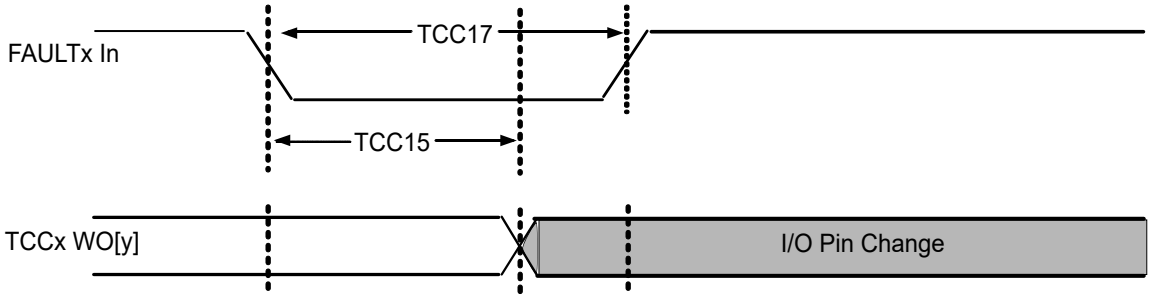



**Table 54-66.** TCCx Timer Capture Module Electrical Specifications (1)

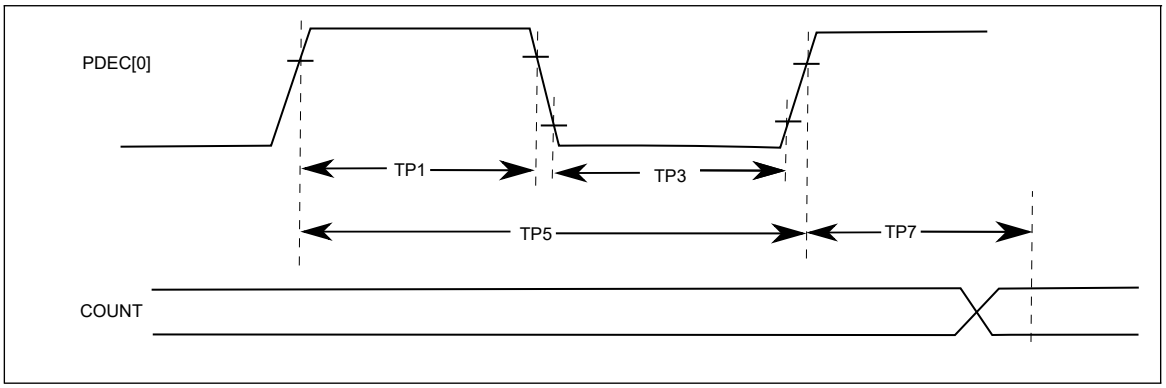
| AC Characteristics | | | Standard Operation Conditions:VDDIO=VDDANA 1.71V to 3.63V(unless otherwise stated) **Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics (1) | Min. | Typical | Max. | Units | Conditions |
| TCC1 | $TCC_{INLOW}$ | Capture TCCx Input Low Time | 2/fGCLK_TCCx | - | - | ns | VDDIOx(min) and meet TCC5 spec |
| TCC3 | $TCC_{INHIGH}$ | Capture TCCx Input High Time | 2/fGCLK_TCCx | - | - | ns | VDDIOx(min) and meet TCC5 spec |
| TCC5 | $TCC_{INPERIOD}$ | Capture Input Period | 4/fGCLK_TCCx | - | - | ns | VDDIOx(min) |
| TCC7 | $TCC_{OUTLOW}$ | Compare TCCx Output Low Time | 1/fGCLK_TCCx | - | - | ns | VDDIOx(min) and meet TCC11 spec |
| TCC9 | $TCC_{OUTHIGH}$ | Compare TCCx Output High Time | 1/fGCLK_TCCx | - | - | ns | VDDIOx(min) and meet TCC11 spec |
| TCC11 | $TCC_{OUTPERIOD}$ | Compare Output Period | 2/fGCLK_TCCx | - | - | ns | VDDIOx(min) |
| TCC13 | fGCLK_TCCx | GCLK for TCCx | - | - | 200 | MHz | |
| TCC15 | $TCC_{FD}$ | Fault Input to I/O Pin Change | - | - | 35 | ns | |
| TCC17 | $TCC_{FLT}$ | Fault Input Pulse Width | 10 | - | - | ns | |

**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

## 54.14.9. PDEC - Position Decoder Electrical Specifications

**Figure 54-28.** Position Decoder (PDEC) Timing Diagrams Counter Mode





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 54-29.** PDEC[0]/PDEC[1] Input Characteristics

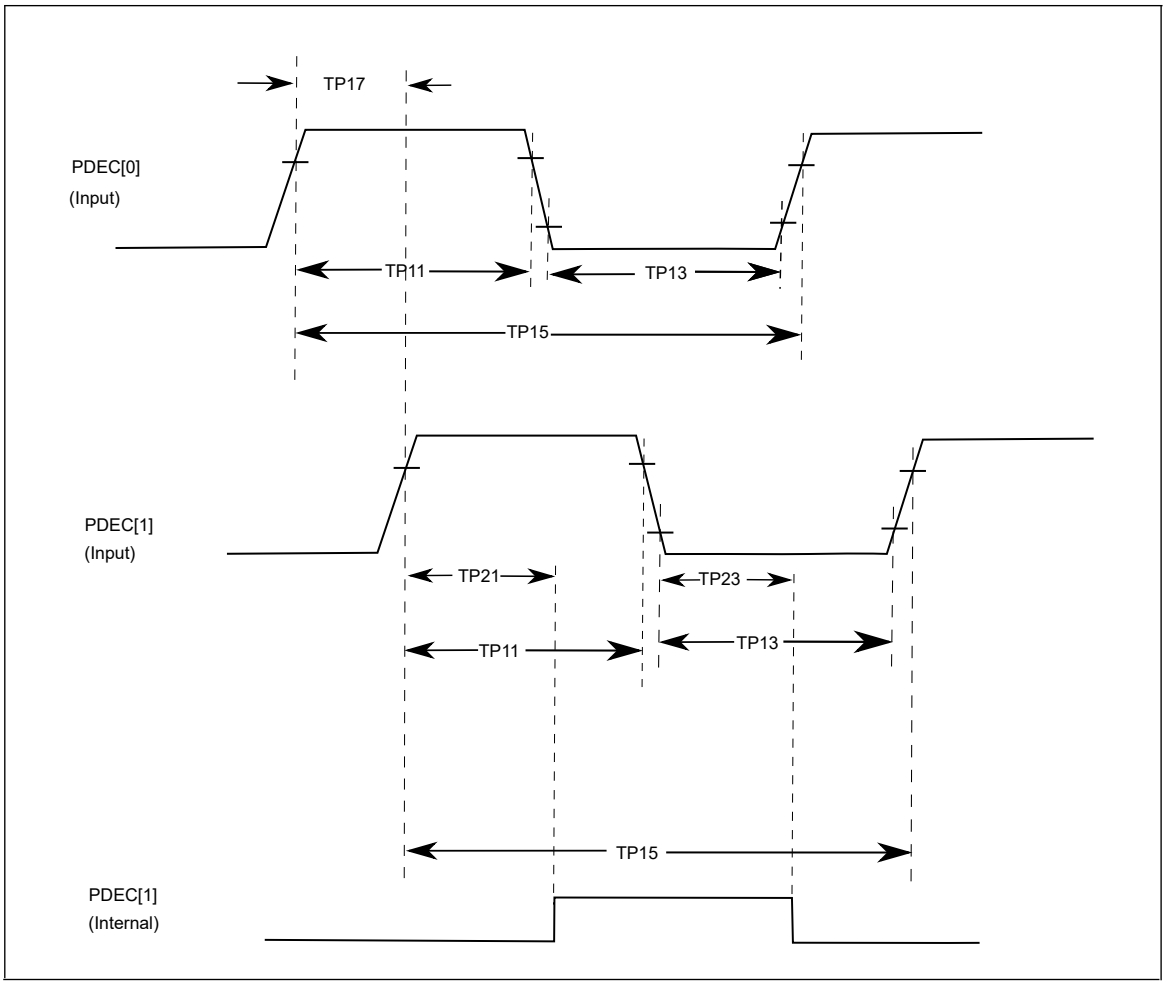



**Table 54-67.** Quadrature Encoder Interface Electrical Specifications (1)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| TP1 | TtPH | TPCK high time | 2/fQEICLK+1.3 | — | | ns | VDDIOx(min) -to- VDDIOx(max) |
| TP3 | TtPL | TPCK low time | 2/fQEICLK + 0.9 | — | | ns | |
| TP5 | TtPP | TPCK input period | 4/fQEICLK+ 2.2 | — | | ns | |
| TP7 | TCKEXTDLY | Delay from External TxCK Clock Edge to counter Increment | | — | 4/fQEICLK+20.5 | ns | |
| TP11 | TPDH | Position Decoder Input High Time | 4/fQEICLK+30.9 | — | | ns | |
| TP13 | TPDL | Position Decoder Input Low Time | 4/fQEICLK+30.9 | — | | ns | |
| TP15 | TPDIN | Position Decoder Input Period | 8/fQEICLK+61.8 | — | | ns | |
| TP17 | TPDP | Position Decoder Phase Period | 2/fQEICLK+15.5 | | | ns | |
| TP21 | TPDFH | Filter Time to Recognize Low, with Digital Filter | 4/fQEICLK+20.5 | — | | ns | |
| TP23 | TPDFL | Filter Time to Recognize High, with Digital Filter | 4/fQEICLK+20.5 | — | | ns | |
| TP24 | fQEICLK | GCLK for PDEC | | — | 200 | MHz | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

## 54.15. USB Characteristics

The USB on-chip buffers comply with the Universal Serial Bus (USB) v2.0 standard. All AC parameters related to these buffers can be found within the USB 2.0 electrical specifications.

The USB interface is USB-IF certified:

- TID 40001782 - Peripheral Silicon > Low/Full Speed > Silicon Building Blocks
- TID 120000724 - Embedded Hosts > Full Speed

Electrical configuration required to be USB-compliant:

- the CPU frequency must be higher than 16 MHz when USB is active (No constraint for USB suspend mode)
- the operating voltages must be 3.3V (Min. 3.0V, Max. 3.6V).
- the GCLK_USB frequency accuracy source must be less than:
  - in USB FS device mode, 48MHz +/-0.25%
  - in USB FS host mode, 48MHz +/-0.05%

**Table 54-68.** GCLK_USB Clock Setup Recommendations

| Clock setup | | USB Device | USB Host |
|---|---|---|---|
| DFLL48M | Open loop | No | No |
| | Close loop, Ref. internal RC OSC source | No | No |
| | Close loop, Ref. external XOSC source | **Yes** | No |
| | Close loop, Ref. SOF (USB recovery mode)(1) | **Yes**(2) | N/A |
| FDPLL | internal RC OSC (32K, 8M...) | No | No |
| | external OSC (<1MHz) | **Yes** | No |
| | external OSC (>1MHz) | **Yes**(3) | **Yes** |

**Notes:**

1. When using DFLL48M in USB recovery mode, the Fine Step value must be 0xA to guarantee a USB clock at +/-0.25% before 11ms after a resume. Only usable in LDO regulator mode.
2. Very high signal quality and crystal-less. It is the best setup for USB Device mode.
3. FDPLL lock time is short when the clock frequency source is high (> 1 MHz). Thus, FDPLL and external OSC can be stopped during USB suspend mode to reduce consumption and guarantee a USB wake-up time (See TDRSMDN in the USB 2.0 specification).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 55. Electrical Characteristics at 105°C

The specifications for 105°C temperature devices are identical to those shown in Electrical Characteristics at 85°C, with the exception of the parameters listed in this chapter.

## 55.1. General Operating Ratings (105°C)

The device must operate with in the ratings listed in the Table 58-2. Thermal Operation Conditions.

## 55.2. Power Consumption (105°C)

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - CPU is running on Flash with automatic wait state
  - Low power cache enabled
  - BOD33 is disabled
  - I/Os are inactive input mode, with input trigger disabled
- **Oscillators**
  - XOSC0 (crystal oscillator) running with external 32 MHz crystal
  - XOSC32K (32 kHz crystal oscillator) running with external 32 kHz crystal in LP mode
  - FDPLL is using XOSC32K as reference on LDO and external clock 32768 on Buck mode
  - DFLL48M is using XOSC32K as reference


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 55-1.** Current Consumption - Active and Idle Mode

| Mode | conditions | Regulator | Clock | VDD | $T_A$ | Typ. | Max | Units |
|---|---|---|---|---|---|---|---|---|
| ACTIVE | COREMARK (1) | LDO | FDPLL 120MHz | 1.8 | Max at 105°C Typ at 25°C | 136 | 191 | µA/MHz |
| | | | | 3.3 | | 137 | 193 | |
| | | | DFLL 48MHz | 1.8 | | 136 | 271 | |
| | | | | 3.3 | | 136 | 272 | |
| | | | XOSC 32MHz | 1.8 | | 146 | 346 | |
| | | | | 3.3 | | 149 | 347 | |
| | | BUCK | FDPLL 120MHz | 1.8 | | 103 | 151 | |
| | | | | 3.3 | | 65 | 133 | |
| | | | DFLL 48MHz | 1.8 | | 102 | 225 | |
| | | | | 3.3 | | 63 | 169 | |
| | | | XOSC 32MHz | 1.8 | | 110 | 283 | |
| | | | | 3.3 | | 73 | 224 | |
| IDLE | NA | LDO | FDPLL 120MHz | 1.8 | | 21 | 78 | |
| | | | | 3.3 | | 23 | 81 | |
| | | | DFLL 48MHz | 1.8 | | 21 | 156 | |
| | | | | 3.3 | | 21 | 156 | |
| | | | XOSC 32MHz | 1.8 | | 25 | 231 | |
| | | | | 3.3 | | 27 | 233 | |
| | | BUCK | FDPLL 120MHz | 1.8 | | 16 | 59 | |
| | | | | 3.3 | | 11 | 54 | |
| | | | DFLL 48MHz | 1.8 | | 16 | 119 | |
| | | | | 3.3 | | 10 | 85 | |
| | | | XOSC 32MHz | 1.8 | | 21 | 180 | |
| | | | | 3.3 | | 19 | 135 | |

**Note:**

1. System Configuration used:
   - MCLK all APB clocks masked except MCLK and NVMCTRL
   - MCLK.AHBMASK = 0x00C00FFF
   - CMCC enabled


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 55-2.** Standby, Hibernate, Backup and OFF Mode Current Consumption

| Mode | Conditions | Regulator Mode | $V_{DD}$ | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| STANDBY | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), no peripheral running<br>No System RAM retained (PM.STDBYCFG.RAMCFG=0x2).<br>8KB backup RAM retained | LDO | 1.8V | Max at 105°C Typ at 25°C | 43 | 3316 | µA |
| | | | 3.3V | | 43 | 3322 | |
| | | BUCK | 1.8V | | 26 | 2211 | |
| | | | 3.3V | | 17 | 1581 | |
| | fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), no peripheral running<br>No System RAM retained (PM.STDBYCFG.RAMCFG=0x2).<br>8KB backup RAM retained | LDO | 1.8V | | 85 | 5106 | |
| | | | 3.3V | | 85 | 5110 | |
| | | BUCK | 1.8V | | 65 | 3907 | |
| | | | 3.3V | | 47 | 2756 | |
| | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), RTC running on XOSC32K<br>No System RAM retained (PM.STDBYCFG.RAMCFG=0x2).<br>8KB backup RAM retained | LDO | 1.8V | | 43 | 3322 | |
| | | | 3.3V | | 44 | 3329 | |
| | | BUCK | 1.8V | | 26 | 2218 | |
| | | | 3.3V | | 18 | 1587 | |
| | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), RTC running on XOSC32K 32KB<br>System RAM retained (PM.STDBYCFG.RAMCFG=0x1).<br>8KB backup RAM retained | LDO | 1.8V | | 45 | 3462 | |
| | | | 3.3V | | 46 | 3469 | |
| | | BUCK | 1.8V | | 27 | 2311 | |
| | | | 3.3V | | 19 | 1652 | |
| | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), RTC running on XOSC32K<br>Full System RAM retained (PM.STDBYCFG.RAMCFG=0x0).<br>8KB backup RAM retained | LDO | 1.8V | | 53 | 3997 | |
| | | | 3.3V | | 53 | 4003 | |
| | | BUCK | 1.8V | | 32 | 2668 | |
| | | | 3.3V | | 22 | 1903 | |
| | fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), no peripheral running<br>Full System RAM retained (PM.STDBYCFG.RAMCFG=0x0).<br>8KB backup RAM retained | LDO | 1.8V | | 101 | 3126 | |
| | | | 3.3V | | 101 | 3140 | |
| | | BUCK | 1.8V | | 78 | 2375 | |
| | | | 3.3V | | 55 | 1659 | |
| | fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), RTC running on XOSC32K<br>Full System RAM retained (PM.STDBYCFG.RAMCFG=0x0).<br>8KB backup RAM retained | LDO | 1.8V | | 102 | 3132 | |
| | | | 3.3V | | 103 | 3146 | |
| | | BUCK | 1.8V | | 79 | 2383 | |
| | | | 3.3V | | 56 | 1666 | |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 55-2.** Standby, Hibernate, Backup and OFF Mode Current Consumption (continued)

| Mode | Conditions | Regulator Mode | $V_{DD}$ | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| HIBERNATE | no peripheral running<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>No backup RAM retained (PM.HIBCFG.BRAMCFG=0x2) | LDO | 1.8V | Max at 105°C Typ at 25°C | 6 | 168 | µA |
| | | | 3.3V | | 6 | 170 | |
| | | BUCK | 1.8V | | 3 | 111 | |
| | | | 3.3V | | 3 | 111 | |
| | RTC is running on XOSC32K<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>No backup RAM retained (PM.HIBCFG.BRAMCFG=0x2) | LDO | 1.8V | | 6 | 169 | |
| | | | 3.3V | | 7 | 172 | |
| | | BUCK | 1.8V | | 3 | 113 | |
| | | | 3.3V | | 3 | 112 | |
| | RTC is running on XOSC32K<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>4KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x1) | LDO | 1.8V | | 7 | 193 | |
| | | | 3.3V | | 8 | 196 | |
| | | BUCK | 1.8V | | 3 | 129 | |
| | | | 3.3V | | 4 | 128 | |
| | RTC is running on XOSC32K<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>8KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x0) | LDO | 1.8V | | 7 | 217 | |
| | | | 3.3V | | 8 | 219 | |
| | | BUCK | 1.8V | | 4 | 144 | |
| | | | 3.3V | | 4 | 143 | |
| | RTC is running on XOSC32K<br>32KB System RAM retained (PM.HIBCFG.RAMCFG=0x1)<br>8KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x0) | LDO | 1.8V | | 9 | 350 | |
| | | | 3.3V | | 10 | 352 | |
| | | BUCK | 1.8V | | 5 | 233 | |
| | | | 3.3V | | 4 | 228 | |
| | RTC is running on XOSC32K<br>Full System RAM retained (PM.HIBCFG.RAMCFG=0x0)<br>8KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x0) | LDO | 1.8V | | 16 | 873 | |
| | | | 3.3V | | 17 | 874 | |
| | | BUCK | 1.8V | | 9 | 580 | |
| | | | 3.3V | | 7 | 431 | |
| BACKUP | powered by VDDIO,<br>no RTC running VDDIO+VDDANA consumption<br>No backup RAM retained (PM.BKUPCFG.BRAMCFG=0x2) | BUCK | 1.8V | Max at 105°C Typ at 25°C | 2.1 | 160 | µA |
| | | | 3.3V | | 2.5 | 162 | |
| | powered by VDDIO with RTC running on<br>XOSC32K VDDIO+VDDANA consumption<br>No backup RAM retained (PM.BKUPCFG.BRAMCFG=0x2) | | 1.8V | | 2.7 | 161 | |
| | | | 3.3V | | 3.3 | 164 | |
| | powered by VDDIO,<br>no RTC running VDDIO+VDDANA consumption<br>4KB backup RAM retained (PM.BKUPCFG.BRAMCFG=0x1) | | 1.8V | | 2.4 | 183 | |
| | | | 3.3V | | 2.8 | 185 | |
| | powered by VDDIO,<br>no RTC running VDDIO+VDDANA consumption<br>8KB backup RAM retained (PM.BKUPCFG.BRAMCFG=0x0) | | 1.8V | | 2.7 | 207 | |
| | | | 3.3V | | 3.1 | 209 | |
| | Battery Backup mode powered by VBAT with RTC running on | | 1.8V | | 2.7 | 161 | |
| | | | 3.3V | | 3.3 | 164 | |
| OFF | - | | 1.8V | | 0.191 | 11 | µA |
| | | | 3.3V | | 0.331 | 13 | |

## 55.3. Analog Characteristics (105°C)

### 55.3.1. Power-On Reset (POR) Characteristics (105°C)

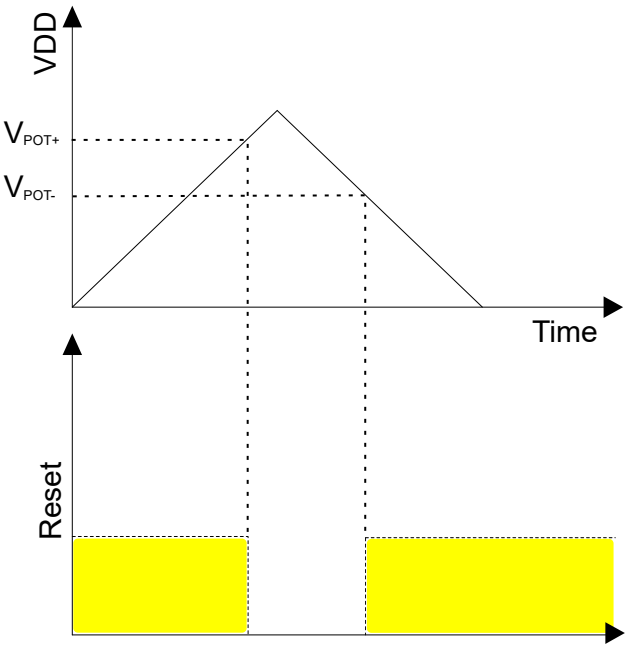
**Table 55-3.** POR Characteristics

| Symbol | Parameters | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| $V_{POT+}$ | Voltage threshold Level on $V_{DDIO}$ rising | 1.52 | - | 1.65 | V |
| $V_{POT-}$ | Voltage threshold Level on $V_{DDIO}$ falling | 0.97 | - | 1.36 | V |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Figure 55-1.** POR Operating Principle




**Note:** The shaded area indicates that the device is in a Reset state.

### 55.3.2. Brown-Out Detectors (BOD) Characteristics (105°C)

**Figure 55-2.** BOD33 Hysteresis OFF

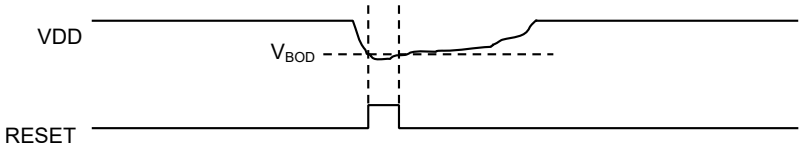



**Figure 55-3.** BOD33 Hysteresis ON

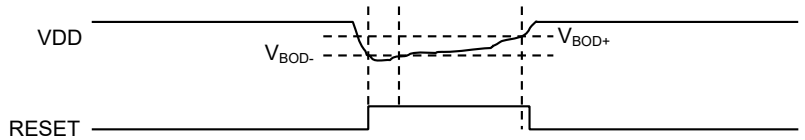



**Table 55-4.** BOD33 Characteristics on VDD and VBAT Monitoring in Normal Mode (During Power-up Phase and Active Mode)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71 V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +105°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min.(1) | Typical | Max.(2) | Units | Conditions |
| REG1 | VBOD33 | VDDANA and VBAT BOD levels | 1.453 | — | 1.565 | V | ActiveLEVEL[7:0] = 0x00 (Min Value) , HYST[3:0] =0x02 |
| | | | 1.598 | — | 1.719 | V | ActiveLEVEL[7:0] = 0x19 (Recommended Value) , HYST[3:0] =0x02 |
| | | | 1.616 | — | 1.737 | V | ActiveLEVEL[7:0] = 0x1C (Fuse Value) , HYST[3:0] =0x02 |
| | | | 2.925 | — | 3.316 | V | ActiveLEVEL[7:0] = 0xFF(3) (Max Value) , HYST[3:0] = 0x00 |
| REG3 | $VBOD33_{LEVEL_STEP}$ | VBOD33 step size, LEVEL[7:0] | — | 6 | — | mV | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 55-4.** BOD33 Characteristics on VDD and VBAT Monitoring in Normal Mode (During Power-up Phase and Active Mode) (continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71 V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +105°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min.(1) | Typical | Max.(2) | Units | Conditions |
| REG4 | $VBOD33_{HYST_STEP}$ | VBOD33 Hysteresis step size, HYST[3:0] | — | 6 | — | mV | |
| REG5 | TSTART(4) | Startup time (Time from enable to RDY) | — | 27 | — | µs | |

**Notes:**

1. BOD- = VBOD(min) = (1.5 + (LEVEL[7:0] * $VBOD33_{LEVEL_STEP}$))
2. BOD+ = VBOD(max) = (VBOD(min) + (HYST[3:0] * $VBOD33_{HYST_STEP}$))
3. If a hysteresis value is programmed, the following equation must be respected to avoid any overflow on VBOD33(max): LEVEL[7:0] value < 255 - HYST[3:0] value.
4. These values are based on design simulation, and not covered by production test limits or characterization.

**Table 55-5.** BOD33 Characteristics on VDD and VBAT Monitoring in Low-Power Mode (During Standby/Backup/Hibernate Modes)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71 V to 3.63V (unless otherwise stated) **Operating temperature -40°C ≤ TA ≤ +105°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min.(1) | Typical | Max.(2) | Units | Conditions |
| REG7 | VBOD33 | VDDANA and VBAT BOD levels | 1.39 | — | 1.63 | V | ActiveLEVEL[7:0] = 0x00 (Min Value) , HYST[3:0] =0x02 |
| | | | 1.52 | — | 1.8 | V | ActiveLEVEL[7:0] = 0x19 (Recommended Value) , HYST[3:0] =0x02 |
| | | | 1.54 | — | 1.82 | V | ActiveLEVEL[7:0] = 0x1C (Fuse Value) , HYST[3:0] =0x02 |
| | | | 2.8 | — | 3.28 | V | ActiveLEVEL[7:0] = 0xFF(3) (Max Value) , HYST[3:0] = 0x00 |
| REG9 | $VBOD33_{LEVEL_STEP}$ | VBOD33 step size, LEVEL[7:0] | — | 6 | — | mV | |
| REG10 | $VBOD33_{HYST_STEP}$ | VBOD33 Hysteresis step size, HYST[3:0] | — | 6 | — | mV | |
| REG11 | TSTART(4) | Startup time (Time from enable to RDY) | — | 27 | — | µs | |

**Notes:**

1. BOD- = VBOD(min) = (1.5 + (LEVEL[7:0] * $VBOD33_{LEVEL_STEP}$))
2. BOD+ = VBOD(max) = (VBOD(min) + (HYST[3:0] * $VBOD33_{HYST_STEP}$))
3. If a hysteresis value is programmed, the following equation must be respected to avoid any overflow on VBOD33(max): LEVEL[7:0] value < 255 - HYST[3:0] value.
4. These are based on design simulation, and not covered by production test limits or characterization.

**Table 55-6.** BOD33 Power Consumption

| Symbol | CPU Mode | Conditions | $T_A$ | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Active / Idle | VDD = 1.8V | Max. 105°C Typ 25°C | 8.52 | 13.26 | µA |
| | | VDD = 3.3V | | 10.10 | 15.70 | |
| | Standby with BOD continuous normal mode | VDD = 1.8V | | 4.71 | 6.74 | |
| | | VDD = 3.3V | | 6.01 | 8.59 | |
| | Standby with BOD continuous low power mode or Hibernate mode | VDD = 1.8V | | 0.15 | 0.25 | |
| | | VDD = 3.3V | | 0.21 | 0.33 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 55.3.3. Analog-to-Digital Converter (ADC) Characteristics (105°C)

**Table 55-7.** ADC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +105°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **Device Supply** | | | | | | | |
| ADC1 | VDDANA | ADC Module Supply | VDDANA(min) | — | VDDANA(max) | V | VDDIOx = VDDANA |
| **Reference Inputs** | | | | | | | |
| ADC3 | VREF | ADC Reference Voltage | 1 | — | VDDANA-0.4 | V | Internal or External Reference VREF ≤ VDDANA - 0.4V |
| **Analog Input Range** | | | | | | | |
| ADC7 | AFS | Full-Scale Analog Input Signal Range (Single-Ended) | GNDANA | — | VREF | V | |
| ADC9 | | Full-Scale Analog Input Signal Range (Differential) | -VREF | — | +VREF | V | |
| ADC11 | VCM | Input common mode voltage | 0 | — | VDDANA | V | CTRLA.R2R =1 |
| | | | See Note 2 | | | | CTRLA.R2R =0 |
| ADC13 | $T_{SETTING}$ | ADC stabilization Time | — | 10 | — | µs | CTRLA.ENABLE=1 or CTRLA.ONEDEMAND=1 |

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. Limit the input common mode voltage using the following equations (where, VCM_IN is the input channelcommon mode voltage):
   When CTRLA.R2R = 0:
   - VCM_IN < 0.75*VREF
   - VCM_IN > Maximum of (0, VREF-VDDANA-0.7, 1.25*VREF-VDDANA)

**Table 55-8.** ADC Conversion Timing Requirements (4)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +105°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **ADC Clock Requirements** | | | | | | | |
| ADC21 | TAD | ADC Clock Period | 62.5 | — | 3125 | ns | |
| ADC23 | fGCLK_ADCx | ADCx Module GCLK max input freq | — | — | 100 | MHz | |
| **ADC Single-Ended Throughput Rates** | | | | | | | |
| ADC25 | FTP (Single-Ended Mode) | Throughput Rate (3) (Single-Ended) | 24.62 | — | 1231 | ksps | 12-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 (1) |
| | | | 26.67 | — | 1333 | | 10-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 (1) |
| | | | 32 | — | 1600 | | 8-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 (1) |
| | | | 20 | — | 1000 | ksps | 12-bit resolution, Rsource ≤6,550Ω SAMPCTRL.SAMPLEN=n/a (2) |
| | | | 21.30 | — | 1067 | | 10-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a (2) |
| | | | 24.62 | — | 1231 | | 8-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a (2) |
| **ADC Differential Mode Throughput Rates** | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 55-8.** ADC Conversion Timing Requirements [4] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +105°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| ADC27 | FTP (Differential Mode) | Throughput Rate [3] (Differential Mode) | 24.62 | — | 1231 | ksps | 12-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 29.09 | — | 1455 | | 10-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 35.56 | — | 1778 | | 8-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 20 | — | 1000 | ksps | 12-bit resolution, Rsource ≤6,550Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 22.86 | — | 1143 | | 10-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 26.67 | — | 1333 | | 8-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |

**Notes:**

1. ADC_ Sample time = ((SAMPCTRL.SAMPLEN + 1) * TAD) and SAMPCTRL.OFFCOMP=0.
2. ADC_ HDW forces sample time to 4*TAD when SAMPCTRL.OFFCOMP=1, user SAMPCTRL.SAMPLEN is ignored.
3. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)). # of active analog channels used =1.
4. These values are based on simulation and not covered by production test limits or characterization.

**Table 55-9.** ADC Sample Timing Requirements [4]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) **Operating temperature: -40°C ≤ TA ≤ +105°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| ADC29 | TSAMP | ADC Sample Time [1,2,3] | 1[1] | — | — | TAD | TAD(min) Ext Analog Input Rsource ≤ 147 Ω |
| | | | 2[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 2,250 Ω |
| | | | 3[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 4,400 Ω |
| | | | 4[1,2] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 6,550 Ω |
| | | | 5[1] | — | — | | TAD(min), Ext Analog Input Rsource ≤ 8,700 Ω |
| | | | 6[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 10,850 Ω |
| | | | See Note 5 | | | ns | Sample time with DAC as input |
| | | | 10000 | — | — | | Sample time with Temp sensor or band gap as input |
| ADC31 | TCNV | Conversion Time [3] (Single-Ended Mode) | 12 | | | TAD | 12-bit resolution |
| | | | 11 | | | | 10-bit resolution |
| | | | 9 | | | | 8-bit resolution |
| ADC33 | | Conversion Time [3] (Differential Mode) | 12 | | | TAD | 12-bit resolution |
| | | | 10 | | | | 10-bit resolution |
| | | | 8 | | | | 8-bit resolution |
| ADC35 | CSAMPLE | ADC Internal Sample Cap | 2 | — | 3 | pf | |
| ADC37 | RSAMPLE | ADC Internal impedance | — | — | 2000 | Ω | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. When SAMPCTRL.OFFCOMP = 0:
   - TSAMP = (((RSAMPLE + RSOURCE) * CSAMPLE * 9.7) / TAD)+1 rounded down to nearest whole integer
   - User SAMPCTRL.SAMPLEN = (TSAMP - 1)
2. When SAMPCTRL.OFFCOMP=1:
   - TSAMP = 4 (Forced by HDW)
   - User SAMPCTRL.SAMPLEN = (n/a, Ignored by HDW)
3. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)).
4. These values are based on simulation and are not covered by production test limits or characterization.
5. See ts specified in DAC Electrical Characteristics.

**Table 55-10.** Differential Mode ADC Electrical Specifications [1]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **DIFFERENTIAL MODE ADC Accuracy** | | | | | | | |
| DADC41 | Res | Resolution | 8 | — | 12 | | Selectable 8, 10, 12 bit Resolution Ranges |
| DADC43 | EN0B | Effective Number of bits | 10.5 | — | — | bits | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| DADC45 | | | 10.5 | — | — | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC47 | INL | Integral Nonlinearity | -1.7 | — | +1.7 | LSb | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| DADC49 | | | -1.5 | — | +1.5 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC51 | DNL | Differential Nonlinearity | -1.0 | — | +1.0 | | 1msps, R2R disabled,VREF = VDDANA = VDDIO = 3.0 |
| DADC53 | | | -1.0 | — | +1.0 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC55 | GERR | Gain Error with REFCTRL.REFCOMP=1 | - 0.21 | — | +0.20 | %FSR | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC57 | | | - 0.12 | — | +0.21 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC59 | | | - 10 | — | +6.7 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC61 | | | - 0.48 | — | +0.75 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC63 | | Gain Error with REFCTRL.REFCOMP=0 | - 0.30 | — | +0.20 | | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC65 | | | - 0.94 | — | +0.70 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC67 | | | - 10 | — | +5.9 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC69 | | | - 1.20 | — | +1.28 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC71 | E0FF | Offset Error with SAMPCTRL.OFFCOMP=1 | - 3.6 | — | +3.1 | mV | 1msps, Internal VREF = VDDANA = VDDIO = 3.0 |
| DADC73 | | | - 3.3 | — | +2.7 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC75 | | | - 3.6 | — | +2.9 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC77 | | | - 3.6 | — | +3.3 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC79 | | Offset Error with SAMPCTRL.OFFCOMP=0 | - 12.3 | — | +12.3 | | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC81 | | | - 12.2 | — | +12.4 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC83 | | | - 14.3 | — | +14.7 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC85 | | | - 13.6 | — | +14.0 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC87 | TUE[3] | Total Unadjusted Error | -5.2 | — | +5.2 | LSb | 1msps, R2R disabled, Internal VREF = VDDANA = VDDIO = 3.0 |
| DADC89 | | | -5.7 | — | +5.7 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| **DIFFERENTIAL MODE ADC Dynamic Performance** | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 55-10.** Differential Mode ADC Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| DADC91 | SINAD | Signal to Noise and Distortion | 65.2 | — | — | dB | VREF=VDDANA=VDDIO=3.0v @ 12bit max sampling rate, Fin = 14 kHz / Full range Input signal [2] |
| DADC93 | SNR | Signal to Noise ratio | 64.6 | — | — | | |
| DADC95 | SFDR | Spurious Free Dynamic Range | 76.6 | — | — | | |
| DADC97 | THD | Total Harmonic Distortion | — | — | -78.6 | | |
| DADC99 | Nrms | Noise RMS | — | — | 2.4 | mV | External VREF = 2.0V, VDDANA = VDDIO = 3.0, Constant Input Voltage |
| | | | | | 2.5 | | VREF = VDDANA = VDDIO = 3.0, Constant Input Voltage |

**Notes:**

1. These values are based on characterization. These values are not covered by test limits in production.
2. All values expressed in decibel refer to the full scale input and are tested with an input signal 0.35dB below full scale; THD measured on the first seven harmonics of the input signal.
3. With REFCTRL.REFCOMP=1 and SAMPCTRL.OFFCOMP=1.

**Table 55-11.** Single Ended Mode ADC Electrical Specifications [1]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **SINGLE ENDED MODE ADC Accuracy** | | | | | | | |
| SADC41 | Res | Resolution | 8 | — | 12 | bits | Selectable 8, 10, 12 bit Resolution Ranges |
| SADC43 | ENOB | Effective Number of bits | 8.9 | — | — | | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| SADC45 | | | 8.85 | — | — | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC47 | INL | Integral Nonlinearity | -3.2 | — | +3.2 | LSb | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| SADC49 | | | -3.0 | — | +3.0 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC51 | DNL | Differential Nonlinearity | -1.0 | — | +1.0 | | 1msps, R2R disabled,VREF = VDDANA = VDDIO = 3.0 |
| SADC53 | | | -1.0 | — | +1.2 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC55 | GERR | Gain Error with REFCTRL.REFCOMP=1 | - 0.3 | — | +0.3 | %FSR | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| SADC57 | | | - 0.16 | — | +0.3 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC59 | | | - 11 | — | +7 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| SADC61 | | | - 0.5 | — | +0.7 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| SADC71 | E0FF | Offset Error with SAMPCTRL.OFFCOMP=1 | - 19 | — | +9.3 | mV | 1msps, Internal VREF = VDDANA = VDDIO = 3.0 |
| SADC73 | | | - 20.7 | — | +17 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC75 | | | - 24 | — | +26 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| SADC77 | | | - 27 | — | +24 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| SADC87 | TUE[3] | Total Unadjusted Error | -18.3 | — | +18.3 | LSb | 1msps, R2R disabled, Internal VREF = VDDANA = VDDIO = 3.0 |
| SADC89 | | | -19.1 | — | +19.1 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| **SINGLE ENDED MODE ADC Dynamic Performance** | | | | | | | |
| SADC91 | SINAD | Signal to Noise and Distortion | 55.7 | — | — | dB | VREF=VDDANA=VDDIO=3.0v @ 12bit max sampling rate, Fin = 14 kHz / Full range Input signal [2] |
| SADC93 | SNR | Signal to Noise ratio | 54.7 | — | — | | |
| SADC95 | SFDR | Spurious Free Dynamic Range | 67.9 | — | — | | |
| SADC97 | THD | Total Harmonic Distortion | — | — | -65.8 | | |
| SADC99 | Nrms | Noise RMS | — | — | 2.1 | mV | External VREF = 2.0V, VDDANA = VDDIO = 3.0, Constant Input Voltage |
| | | | | | 1.7 | | VREF = VDDANA = VDDIO = 3.0, Constant Input Voltage |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. These values are based on characterization. These values are not covered by test limits in production.
2. All values expressed in decibel refer to the full scale input and are tested with an input signal 0.35dB below full scale; THD measured on the first seven harmonics of the input signal.
3. With REFCTRL.REFCOMP=1 and SAMPCTRL.OFFCOMP=1.

**Table 55-12.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ $V_{DDANA}$ | Differential mode | fs = 1 Msps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | Max 105°C Typ 25°C | 279 | 326 | µA |
| | | fs = 1 Msps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 482 | 686 | |
| | | fs = 10 ksps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 28 | 85 | |
| | | fs = 10 ksps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 241 | 435 | |
| | Single Ended mode | fs = 1 Msps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | Max 105°C Typ 25°C | 307 | 361 | µA |
| | | fs = 1 Msps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 499 | 730 | |
| | | fs = 10 ksps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 38 | 126 | |
| | | fs = 10 ksps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 245 | 448 | |

## 55.3.4. Digital to Analog Converter (DAC) Characteristics (105°C)

**Table 55-13.** Differential Mode (1)

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| INL | Integral Non Linearity, Best Fit curve from 0x080 to 0xF7F | i12clk=12 MHz VDDANA = 3.0V - External Ref = 2.0V Cload = 50pF | - | ±2.4 | ±3.4 | LSB |
| | | i12clk=12 MHz VDDANA = 3.0V - 1V Internal Ref Cload = 50pF | - | ±3.2 | ±4.2 | |
| DNL | Differential Non Linearity, Best Fit curve from 0x080 to 0xF7F | i12clk=12 MHz VDDANA = 3.0V - External Ref = 2.0V Cload = 50pF | - | ±2.4 | ±3.6 | |
| | | i12clk=12 MHz VDDANA = 3.0V - 1V Internal Ref Cload = 50pF | - | ±3.5 | ±5.4 | |
| Gerr | Gain Error | External Reference voltage | - | ±0.4 | ±1.7 | % FSR |
| | | 1.0V Internal Reference voltage | - | ±0.8 | ±8.5 | |
| Offerr | Offset Error | External Reference voltage | - | ±13 | ±44 | mV |
| | | 1.0V Internal Reference voltage | - | ±8 | ±64 | |
| ENOB | Effective Number Of Bits | Fs = 1 Ms/s - External Ref - CCTRL = 0x2 | 9.7 | 10.7 | 11.0 | Bits |
| SNR | Signal to Noise ratio | | 61.3 | 68.6 | 74.5 | dB |
| THD | Total Harmonic Distortion | | -82.3 | -72.5 | -58.9 | dB |

**Note:**

1. These values are based on characterization. These values are not covered by test limits in production.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 55-14.** Single-Ended Mode [1]

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| INL | Integral Non Linearity, Best Fit curve from 0x080 to 0xF7F | i12clk=12 MHz<br>VDDANA = 3.0V - External Ref = 2.0V<br>Cload = 50pF | - | ±2.7 | ±4.0 | LSB |
| | | i12clk=12 MHz<br>VDDANA = 3.0V - 1V Internal Ref<br>Cload = 50pF | - | ±5.2 | ±8.7 | |
| DNL | Differential Non Linearity, Best Fit curve from 0x080 to 0xF7F | i12clk=12 MHz<br>VDDANA = 3.0V - External Ref = 2.0V<br>Cload = 50pF | - | ±3.5 | ±6.1 | |
| | | i12clk=12 MHz<br>VDDANA = 3.0V - 1V Internal Ref<br>Cload = 50pF | - | ±6.4 | ±9.4 | |
| Gerr | Gain Error | External Reference voltage | - | ±0.3 | ±1.6 | % FSR |
| | | 1.0V Internal Reference voltage | - | ±0.8 | ±8.6 | |
| Offerr | Offset Error | External Reference voltage | - | ±7 | ±25 | mV |
| | | 1.0V Internal Reference voltage | - | ±2 | ±19 | |
| ENOB | Effective Number of Bits | Fs = 1 Ms/s - External Ref - CCTRL = 0x2 | 9.1 | 10.3 | 10.8 | Bits |
| SNR | Signal to Noise Ratio | | 63.5 | 68.6 | 74.5 | dB |
| THD | Total Harmonic Distortion | | -82.3 | -72.8 | -61.0 | dB |

**Note:**

1. These values are based on characterization. These values are not covered by test limits in production.

**Table 55-15.** Power Consumption

| Symbol | Parameters | Conditions | TA | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $I_{DDANA}$ | Differential Mode, DC supply current, 2 output channels - without load | fs = 1 Msps, CCTR L= 0x2, $V_{REF}$ > 2.4V, $V_{DD}$ = 3.3V | Max. 105°C<br>Typ. 25°C | - | 384 | 593 | µA |
| | | fs = 10 ksps, CCTRL = 0x0, $V_{REF}$ < 2.4V, $V_{DD}$= 3.3V | | - | 283 | 457 | |
| | Single-Ended Mode, DC supply current, 2 output channels - without load | fs = 1 Msps, CCTRL = 0x2, $V_{REF}$ > 2.4V, $V_{DD}$ = 3.3V | | - | 306 | 493 | µA |
| | | fs = 10 ksps, CCTRL = 0x0, $V_{REF}$ < 2.4V, $V_{DD}$ = 3.3V | | - | 230 | 369 | |

## 55.3.5. Analog Comparator (AC) Characteristics (105°C)

**Table 55-16.** Analog Comparator Characteristics

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| PNIVR | Positive and Negative input range voltage | - | 0 | - | VDDANA | V |
| ICMR | Input common mode range | - | 0 | - | VDDANA-0.2 | V |
| Off [1] | Offset | High-speed<br>COMPCTRLn.SPEED = 0x3 | -22 | +/-3 | 22 | mV |
| Vhys [2] | Hysteresis | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x0 (HYST25) | 13 | 25 | 46 | mV |
| | | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x1 (HYST50) | 30 | 50 | 78 | |
| | | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x2 (HYST75) | 39 | 75 | 111 | |
| | | High-speed<br>COMPCTRLn.SPEED = 0x3<br>COMPCTRLn.HYST = 0x3 (HYST100) | 64 | 100 | 145 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 55-16.** Analog Comparator Characteristics (continued)

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| Tpd (3) | Propagation Delay<br>Vcm = Vddana/2, Vin = +-100mV overdrive from Vcm | High-speed<br>COMPCTRLn.SPEED = 0x3 | - | 24.1 | 42 | ns |
| Tstart | Startup time | High-speed<br>COMPCTRLn.SPEED = 0x3 | - | 4.7 | 8 | µs |

**Notes:**

1. Hysteresis is disabled.
2. Hysteresis is enabled.
3. Tpd is measured from Vin transition to ACOUT (AC direct output) toggle.

**Table 55-17.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $I_{DDANA}$ | Current consumption for<br>One AC enabled,<br>Hysteresis disabled<br>voltage scaler disabled | COMPCTRLn.SPEED=0x3, $V_{DDANA}$=3.3V | Max.105°C<br>Typ.25°C | 59 | 103 | µA |
| | Current consumption Voltage Scaler only | $V_{DDANA}$=3.3V | | 11 | 18 | |

### 55.3.6. PTC Characteristics

The values in the following Power Consumption table are measured values of power consumption under the following conditions:

**Operating Conditions:**

VDD = 3.0V

**Clocks**

DFLL48M used as main clock source, running undivided at 48 MHz

CPU is running on flash with 2 wait states, at 48 MHz

PTC running at 4 MHz

**PTC Configuration**

Mutual-capacitance mode

One touch channel

**System Configuration**

Standby Sleep mode enabled

RTC running on ULP32K: used to define the PTC scan rate, through the event system

RTC interrupts (wakeup) the CPU to perform PTC scans


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 55-18.** Power Consumption [1]

| Symbol | Parameters | PTC scan rate (msec) | Oversamples | TA | Typ. | Max | Units |
|---|---|---|---|---|---|---|---|
| IDD | Current Consumption | 10 | 4 | Max 105°C Typ 25°C | 137 | 2174 | µA |
| | | | 16 | | 146 | 2194 | |
| | | 50 | 4 | | 77 | 2098 | |
| | | | 16 | | 79 | 2104 | |
| | | 100 | 4 | | 68 | 2092 | |
| | | | 16 | | 69 | 2095 | |
| | | 200 | 4 | | 64 | 2086 | |
| | | | 16 | | 65 | 2089 | |

**Note:**

1. These are based on characterization.

## 55.4. NVM Characteristics

**Table 55-19.** NVM Flash Read Wait States for Worst Case Conditions

| Wait States (WS) | 0 WS | 1 WS | 2 WS | 3 WS | 4 WS | 5 WS | 6 WS | Auto WS |
|---|---|---|---|---|---|---|---|---|
| Read Operations | 1 cycle | 2 cycles | 3 cycles | 4 cycles | 5 cycles | 6 cycles | 7 cycles | n cycles |
| CPU FMax (MHz)[1] | 19 | 38 | 57 | 76 | 95 | 100 | 120 | 120 |

**Note:**

1. VDD > 1.71.

Maximum operating frequencies are given in the table above, but are limited by the Embedded Flash access time when the processor is fetching code out of it. Theses tables provide the device maximum operating frequency defined by the field RWS of the NVMCTRL CTRLA register when automatic wait states (AUTOWS) is disabled. This field defines the number of Wait states required to access the Embedded Flash Memory.

## 55.5. Oscillators Characteristics (105°C)

### 55.5.1. Crystal Oscillator (XOSC) Characteristics (105°C)

**Table 55-20.** Multi Crystal Oscillator Electrical Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| Tstart | Startup time | F = 8 MHz - CL=20 pF - Cshunt = 2 pF - IMULT = 0x3 | - | 39700 | 72200 | Cycles |
| | | F = 16 MHz - CL=20 pF - Cshunt = 1,5 pF - IMULT = 0x4 | - | 37550 | 73000 | |
| | | F = 24 MHz - CL=20 pF - Cshunt = 2,5 pF - IMULT = 0x5 | - | 32700 | 68500 | |
| | | F = 48 MHz - CL=13 pF - Cshunt = 5 pF - IMULT = 0x6 | - | 18400 | 38500 | |

**Table 55-21.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | F = 8 MHz - $C_L$ = 20 pF - IMULT = 0x3, ENALC = OFF | Max. 105°C, Typ. 25°C | 0.43 | 1.56 | mA |
| | | ENALC = ON | | 0.16 | 1.16 | |
| | | F = 16 MHz - $C_L$ = 20 pF - IMULT = 0x5, ENALC = OFF | | 1.31 | 3.23 | |
| | | ENALC = ON | | 0.25 | 1.33 | |
| | | F = 32 MHz - $C_L$ = 13 pF - IMULT = 0x5, ENALC = OFF | | 2.92 | 5.74 | |
| | | ENALC = ON | | 0.40 | 1.92 | |
| | | F = 48 MHz - $C_L$ = 13 pF - IMULT = 0x6, ENALC = OFF | | 2.70 | 5.82 | |
| | | ENALC = ON | | 0.76 | 2.64 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 55.5.2. External 32 kHz Crystal Oscillator (XOSC32K) Characteristics (105°C)

**Table 55-22.** Power Consumption

| Symbol | Parameter | Conditions | TA | Gain Mode | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current consumption | VDD=3.0V | Max 105°C Typ 25°C | Std. | 1.5 | 2.5 | µA |
| | | | | High | 1.9 | 3.3 | |

### 55.5.3. Internal Ultra-Low Power 32 kHz RC Oscillator (OSCULP32K) Characteristics (105°C)

**Table 55-23.** Ultra-Low Power Internal 32 kHz RC Oscillator Electrical Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $F_{OUT}$ | Output frequency | [-40, +105]°C, VDDANA>1.71V | 26.00 | 32.768 | 39.50 | kHz |

### 55.5.4. Digital Frequency Locked Loop (DFLL48M) Characteristics (105°C)

**Table 55-24.** DFLL48M Characteristics - Open Loop Mode (1)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{OpenOUT}$ | Output frequency | DFLLVAL after Reset<br>LDO Regulator mode, [-40, 105]°C | 45.57 | 48 | 50.09 | MHz |
| | | DFLLVAL after Reset<br>LDO Regulator mode, [0, 60]°C | 47.12 | 48 | 48.9 | |
| $T_{OpenSTARTUP}$ | Startup time | DFLLVAL after Reset<br>$F_{OUT}$ within 90% of final value | - | 4.3 | 6.5 | µs |

**Note:**

1. DFLL48 in open loop can be used only with LDO regulator.

**Table 55-25.** DFLL48M Power Consumption

| Symbol | Parameter | Conditions | TA | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | Open Loop mode - DFLLVAL after reset VDD = 3.3V | Max. 105°C Typ. 25°C | - | 400 | 1400 | µA |
| | | Closed Loop mode - $f_{REF}$ = 32 .768 kHz VDD = 3.3V | | - | 404 | 1390 | µA |

### 55.5.5. Fractional Digital Phase Lock Loop (FDPLL) Characteristics (105°C)

**Table 55-26.** Fractional Digital Phase Lock Loop Characteristics (2)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| Jp | Period jitter (Peak-Peak value) | $f_{IN}$ = 32 kHz, $f_{OUT}$ = 96 MHz | - | 1.9 | 2.9 | % |
| | | $f_{IN}$ = 32 kHz, $f_{OUT}$ = 200 MHz | - | 3.4 | 5.6 | |
| | | $f_{IN}$ = 3.2 MHz, $f_{OUT}$ = 96 MHz | - | 2.0 | 3.1 | |
| | | $f_{IN}$ = 3.2 MHz, $f_{OUT}$ = 200 MHz | - | 4.3 | 7.1 | |
| Duty (1) | Duty cycle | - | - | 50 | - | % |

**Notes:**

1. These are based on simulation. These values are not covered by test or characterization.
2. These FDPLL200Mn characteristics are applicable with LDO regulator and a direct reference (i.e., REFCLK is XOSC or XOSC32K, not GCLK).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 55-27.** Fractional Digital Phase Lock Loop Power Consumption

| Symbol | Parameter | Conditions | TA | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | Ck = 96 MHz, VDD = 3.3V | Max. 105°C Typ. 25°C | 0.9 | 1.8 | mA |
| | | Ck = 200 MHz, VDD = 3.3V | | 2.0 | 2.6 | |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 56. Electrical Characteristics at 125°C

The specifications for 125°C temperature devices are identical to those shown in Electrical Characteristics at 85°C, with the exception of the parameters listed in this chapter.

## 56.1. General Operating Ratings (125°C)

The device must operate with in the ratings listed in the Table 58-2. Thermal Operation Conditions.

## 56.2. Injection Current (125°C)

Stresses beyond those listed in the table below may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**Table 56-1.** Injection Current(1, 2)

| Symbol | Description | min | Typ. | max | Unit | Comments |
|---|---|---|---|---|---|---|
| $I_{ICL}$ | Input Low Injection Current | -15 | - | - | mA | This Parameter applies to all pins. (1, 3, 4) |
| $I_{ICH}$ | Input High Injection Current | - | - | 15 | mA | This parameter applies to all I/O pins. (2, 3, 4) |
| $\sum I_{ICT}$ | Total Input Injection Current (Sum of all I/O and control pins) Absolute value of $\lvert \sum I_{ICT} \rvert$ | - | - | 18 | mA | Absolute instantaneous sum of all ± input injection currents from all I/O pins.( $\lvert I_{ICL} \rvert + \lvert I_{ICH} \rvert$) $\leq \sum I_{ICT}$ |

**Notes:**

1. VIL source < (VSS - 0.6). Characterized but not tested.
2. VIH source > (VDDIO + 0.6) for all I/O pins.
3. Injection currents > | 0 | can affect the ADC results by approximately 4 to 6 counts (i.e., VIH Source > (VDDIO + 0.6) or VIL source < (GND - 0.6)).
4. Any number and/or combination of I/O pins not excluded under $I_{ICL}$ or $I_{ICH}$ conditions are permitted provided the "absolute instantaneous" sum of the input injection currents from all pins do not exceed the specified $\sum I_{ICT}$ limit. To limit the injection current the user must insert a resistor in series RS between input source voltage and device pin. The resistor value is calculated according to:
   - For negative Input voltages less than (GND-0.6): RS ≥ (((GND - 0.6) - VIL source) / $I_{ICL}$)
   - For positive input voltages greater than (VDDIO+0.6): RS ≥ ((VIH source - (VDDIO+0.6))/ $I_{ICH}$)
   - For Vpin voltages > VDD and < GND then RS = the larger of the values calculated above

## 56.3. Maximum Clock Frequencies (125°C)

**Table 56-2.** Maximum Peripheral Clock Frequencies(1)

| Symbol | Description | Max. | Units |
|---|---|---|---|
| $f_{CPU}$ | CPU clock frequency | 100 | MHz |
| $f_{AHB}$ | AHB clock frequency | 100 | MHz |
| $f_{APBx,\ x = \{A, B, C, D\}}$ | APBA, APBB, APBC and APBD clock frequency | 100 | MHz |
| $f_{GCLK_EIC}$ | EIC input clock frequency | 90 | MHz |
| $f_{GCLK_FREQM_MSR}$ | FREQM Measure | 180 | MHz |
| $f_{GCLK_FREQM_REF}$ | FREQM Reference | 90 | MHz |
| $f_{GCLK_EVSYS_CHANNEL_x,\ x = \{0,..,\ 11\}}$ | EVSYS channel 'x' input clock frequency | 90 | MHz |
| $f_{GCLK_SERCOMx_CORE,\ x = \{0,\ ...\ ,\ 7\}}$ | SERCOMx input clock frequency | 90 | MHz |
| $f_{GCLK_CANx,\ x = \{0,\ 1\}}$ | CANx input clock frequency | 90 | MHz |
| $f_{GCLK_I2S}$ | $I^2S$ input clock frequency | 90 | MHz |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-2.** Maximum Peripheral Clock Frequencies[1] (continued)

| Symbol | Description | Max. | Units |
|---|---|---|---|
| $f_{GCLK_SDHCx_CORE, x = \{0, 1\}}$ | SDHCx input clock frequency | 150 | MHz |
| $f_{GCLK_TCCx, x = \{0, \ldots, 4\}}$ | TCCx input clock frequency | 180 | MHz |
| $f_{GCLK_TCx, x = \{0, \ldots, 3\}}$ | TC0, TC1, TC2, TC3 input clock frequency | 180 | MHz |
| $f_{GCLK_PDEC}$ | PDEC input clock frequency | 180 | MHz |
| $f_{GCLK_CCL}$ | CCL input clock frequency | 90 | MHz |
| $f_{GCLK_CM4_TRACE}$ | CM4 Trace input clock frequency | 100 | MHz |
| $f_{GCLK_AC}$ | AC digital input clock frequency | 90 | MHz |
| $f_{GCLK_ADCx, x = \{0, 1\}}$ | ADCx input clock frequency | 90 | MHz |
| $f_{GCLK_DAC}$ | DAC input clock frequency | 90 | MHz |

**Note:**

1. These values are based on simulation. They are not covered by production test limits or characterization.

## 56.4. Power Consumption (125°C)

The values in this section are measured values of power consumption under the following conditions, except where noted:

- **Operating Conditions**
  - CPU is running on Flash with automatic wait state
  - Low-power cache enabled
  - BOD33 is disabled
  - I/Os are inactive input mode with input trigger disabled
- **Oscillators**
  - XOSC0 (crystal oscillator) running with external 32 MHz crystal
  - XOSC32K (32 kHz crystal oscillator) running with external 32 kHz crystal in LP mode
  - FDPLL is using XOSC32K as reference on LDO and external clock 32768 on Buck mode
  - DFLL48M is using XOSC32K as reference


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 56-3.** Current Consumption - Active and Idle Mode

| Mode | conditions | Regulator | Clock | VDD | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|---|
| ACTIVE | COREMARK (1) | LDO | FDPLL 100 MHz | 1.8 | Max. at 125°C Typ at 25°C | 136 | 229 | µA/MHz |
| | | | | 3.3 | | 137 | 232 | |
| | | | DFLL 48 MHz | 1.8 | | 136 | 370 | |
| | | | | 3.3 | | 136 | 371 | |
| | | | XOSC 32 MHz | 1.8 | | 146 | 611 | |
| | | | | 3.3 | | 149 | 613 | |
| | | BUCK | FDPLL 120 MHz | 1.8 | | 103 | 215 | |
| | | | | 3.3 | | 65 | 176 | |
| | | | DFLL 48 MHz | 1.8 | | 102 | 324 | |
| | | | | 3.3 | | 63 | 242 | |
| | | | XOSC 32 MHz | 1.8 | | 110 | 505 | |
| | | | | 3.3 | | 73 | 370 | |
| IDLE | NA | LDO | FDPLL 100 MHz | 1.8 | | 21 | 114 | |
| | | | | 3.3 | | 23 | 116 | |
| | | | DFLL 48 MHz | 1.8 | | 21 | 252 | |
| | | | | 3.3 | | 21 | 252 | |
| | | | XOSC 32 MHz | 1.8 | | 25 | 367 | |
| | | | | 3.3 | | 27 | 371 | |
| | | BUCK | FDPLL 100 MHz | 1.8 | | 16 | 89 | |
| | | | | 3.3 | | 11 | 78 | |
| | | | DFLL 48 MHz | 1.8 | | 16 | 194 | |
| | | | | 3.3 | | 10 | 147 | |
| | | | XOSC 32 MHz | 1.8 | | 21 | 287 | |
| | | | | 3.3 | | 19 | 223 | |

**Note:**

1. System Configuration used:
   - MCLK all APB clocks masked except MCLK and NVMCTRL
   - MCLK.AHBMASK = 0x00C00FFF
   - CMCC enabled


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 56-4.** Standby, Hibernate, Backup and Off Mode Current Consumption

| Mode | Conditions | Regulator Mode | VDD | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| STANDBY | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), no peripheral running<br>No System RAM retained (PM.STDBYCFG.RAMCFG=0x2).<br>8KB backup RAM retained | LDO | 1.8V | Max at 125°C Typ at 25°C | 43 | 5834 | µA |
| | | | 3.3V | | 43 | 5851 | |
| | | BUCK | 1.8V | | 26 | 3950 | |
| | | | 3.3V | | 17 | 2817 | |
| | fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), no peripheral running<br>No System RAM retained (PM.STDBYCFG.RAMCFG=0x2).<br>8KB backup RAM retained | LDO | 1.8V | | 85 | 8707 | |
| | | | 3.3V | | 85 | 8724 | |
| | | BUCK | 1.8V | | 65 | 6766 | |
| | | | 3.3V | | 47 | 5269 | |
| | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), RTC running on XOSC32K<br>No System RAM retained (PM.STDBYCFG.RAMCFG=0x2).<br>8KB backup RAM retained | LDO | 1.8V | | 43 | 5843 | |
| | | | 3.3V | | 44 | 5860 | |
| | | BUCK | 1.8V | | 26 | 3965 | |
| | | | 3.3V | | 18 | 2891 | |
| | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), RTC running on XOSC32K<br>32KB System RAM retained (PM.STDBYCFG.RAMCFG=0x1).<br>8KB backup RAM retained | LDO | 1.8V | | 45 | 6085 | |
| | | | 3.3V | | 46 | 6102 | |
| | | BUCK | 1.8V | | 27 | 4137 | |
| | | | 3.3V | | 19 | 3012 | |
| | fast wake-up disabled (PM.STDBYCFG.FASTWKUP=0x0), RTC running on XOSC32K<br>Full System RAM retained (PM.STDBYCFG.RAMCFG=0x0).<br>8KB backup RAM retained | LDO | 1.8V | | 53 | 7049 | |
| | | | 3.3V | | 53 | 7068 | |
| | | BUCK | 1.8V | | 32 | 4923 | |
| | | | 3.3V | | 22 | 3479 | |
| | fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), no peripheral running<br>Full System RAM retained (PM.STDBYCFG.RAMCFG=0x0).<br>8KB backup RAM retained | LDO | 1.8V | | 101 | 5543 | |
| | | | 3.3V | | 101 | 5571 | |
| | | BUCK | 1.8V | | 78 | 4266 | |
| | | | 3.3V | | 55 | 3075 | |
| | fast wake-up enabled (PM.STDBYCFG.FASTWKUP=0x3), RTC running on XOSC32K<br>Full System RAM retained (PM.STDBYCFG.RAMCFG=0x0).<br>8KB backup RAM retained | LDO | 1.8V | | 102 | 5563 | |
| | | | 3.3V | | 103 | 5588 | |
| | | BUCK | 1.8V | | 79 | 4270 | |
| | | | 3.3V | | 56 | 3080 | |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 56-4.** Standby, Hibernate, Backup and Off Mode Current Consumption (continued)

| Mode | Conditions | Regulator Mode | VDD | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| HIBERNATE | no peripheral running<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>No backup RAM retained (PM.HIBCFG.BRAMCFG=0x2) | LDO | 1.8V | Max at 125°C Typ at 25°C | 6 | 316 | µA |
| | | | 3.3V | | 6 | 320 | |
| | | BUCK | 1.8V | | 3 | 216 | |
| | | | 3.3V | | 3 | 221 | |
| | RTC is running on XOSC32K<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>No backup RAM retained (PM.HIBCFG.BRAMCFG=0x2) | LDO | 1.8V | | 6 | 318 | |
| | | | 3.3V | | 7 | 322 | |
| | | BUCK | 1.8V | | 3 | 217 | |
| | | | 3.3V | | 3 | 223 | |
| | RTC is running on XOSC32K<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>4 KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x1) | LDO | 1.8V | | 7 | 359 | |
| | | | 3.3V | | 8 | 364 | |
| | | BUCK | 1.8V | | 3 | 244 | |
| | | | 3.3V | | 4 | 250 | |
| | RTC is running on XOSC32K<br>No System RAM retained (PM.HIBCFG.RAMCFG=0x2)<br>8 KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x0) | LDO | 1.8V | | 7 | 400 | |
| | | | 3.3V | | 8 | 405 | |
| | | BUCK | 1.8V | | 4 | 273 | |
| | | | 3.3V | | 4 | 279 | |
| | RTC is running on XOSC32K<br>32 KB System RAM retained (PM.HIBCFG.RAMCFG=0x1)<br>8KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x0) | LDO | 1.8V | | 9 | 636 | |
| | | | 3.3V | | 10 | 641 | |
| | | BUCK | 1.8V | | 5 | 430 | |
| | | | 3.3V | | 4 | 434 | |
| | RTC is running on XOSC32K<br>Full System RAM retained (PM.HIBCFG.RAMCFG=0x0)<br>8 KB backup RAM retained (PM.HIBCFG.BRAMCFG=0x0) | LDO | 1.8V | | 16 | 1574 | |
| | | | 3.3V | | 17 | 1578 | |
| | | BUCK | 1.8V | | 9 | 1061 | |
| | | | 3.3V | | 7 | 1084 | |
| BACKUP | powered by VDDIO,<br>no RTC running VDDIO+VDDANA consumption<br>No backup RAM retained (PM.BKUPCFG.BRAMCFG=0x2) | | 1.8V | Max at 125°C Typ at 25°C | 2.1 | 303.5 | µA |
| | | | 3.3V | | 2.5 | 307.8 | |
| | powered by VDDIO with<br>RTC running on XOSC32K VDDIO+VDDANA consumption<br>No backup RAM retained (PM.BKUPCFG.BRAMCFG=0x2) | | 1.8V | | 2.7 | 304.8 | |
| | | | 3.3V | | 3.3 | 309.5 | |
| | powered by VDDIO,<br>no RTC running VDDIO+VDDANA consumption<br>4 KB backup RAM retained (PM.BKUPCFG.BRAMCFG=0x1) | | 1.8V | | 2.4 | 344.6 | |
| | | | 3.3V | | 2.8 | 348.7 | |
| | powered by VDDIO,<br>no RTC running VDDIO+VDDANA consumption<br>8 KB backup RAM retained (PM.BKUPCFG.BRAMCFG=0x0) | | 1.8V | | 2.7 | 385.1 | |
| | | | 3.3V | | 3.1 | 389.6 | |
| | Battery backup mode powered by VBAT with RTC running on XOSC32K VBAT consumption<br>No backup RAM retained (PM.BKUPCFG.BRAMCFG=0x2), BOD33 enabled in sampled mode PSEL prescaler set to 0x7 (div 256) | | 1.8V | | 2.7 | 305 | |
| | | | 3.3V | | 3.3 | 310 | |
| OFF | - | | 1.8V | | 0.191 | 26.35 | |
| | | | 3.3V | | 0.331 | 31.07 | |

# 56.5. Analog Characteristics (125°C)

## 56.5.1. Power-On Reset (POR) Characteristics (125°C)

**Table 56-5.** POR Characteristics

| Symbol | Parameters | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| $V_{POT+}$ | Voltage threshold Level on $V_{DDIO}$ rising | 1.52 | - | 1.65 | V |

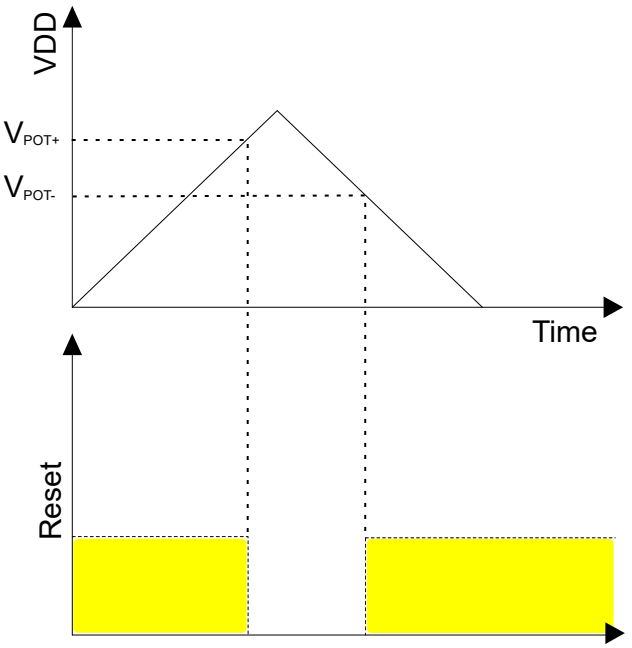

© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 56-5.** POR Characteristics (continued)

| Symbol | Parameters | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| $V_{POT-}$ | Voltage threshold Level on $V_{DDIO}$ falling | 0.97 | - | 1.36 | V |

**Figure 56-1.** POR Operating Principle




**Note:** The shaded area indicates that the device is in a Reset state.

### 56.5.2. Brown-Out Detectors (BOD) Characteristics (125°C)

**Figure 56-2.** BOD33 Hysteresis OFF

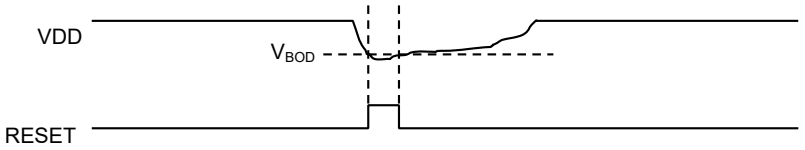
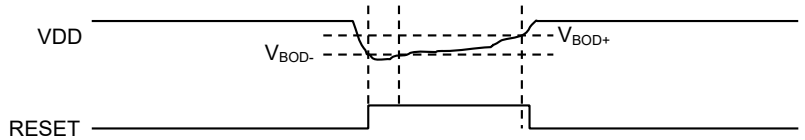



**Figure 56-3.** BOD33 Hysteresis ON





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-6.** BOD33 Characteristics on $V_{DD}$ and $V_{BAT}$ Monitoring in Normal Mode (During Power-up Phase and Active Mode)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71 V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min.(1) | Typical | Max.(2) | Units | Conditions |
| REG1 | VBOD33 | VDDANA and VBAT BOD levels | 1.45 | — | 1.57 | V | ActiveLEVEL[7:0] = 0x00 (Min Value) , HYST[3:0] = 0x02 |
| | | | 1.60 | — | 1.72 | V | ActiveLEVEL[7:0] = 0x19 (Recommended Value) , HYST[3:0] = 0x02 |
| | | | 1.62 | — | 1.74 | V | ActiveLEVEL[7:0] = 0x1C (Fuse Value) , HYST[3:0] = 0x02 |
| | | | 2.93 | — | 3.32 | V | ActiveLEVEL[7:0] = 0xFF(3) (Max Value) , HYST[3:0] = 0x00 |
| REG3 | $VBOD33_{LEVEL_STEP}$ | VBOD33 step size, LEVEL[7:0] | — | 6 | — | mV | |
| REG4 | $VBOD33_{HYST_STEP}$ | VBOD33 Hysteresis step size, HYST[3:0] | — | 6 | — | mV | |
| REG5 | TSTART(4) | Startup time (Time from enable to RDY) | — | 27 | — | µs | |

**Notes:**

1. BOD- = VBOD(min) = (1.5 + (LEVEL[7:0] * $VBOD33_{LEVEL_STEP}$)).
2. BOD+ = VBOD(max) = (VBOD(min) + (HYST[3:0] * $VBOD33_{HYST_STEP}$)).
3. If a hysteresis value is programmed, the following equation must be respected to avoid any overflow on VBOD33(max): LEVEL[7:0] value < 255 - HYST[3:0] value.
4. These are based on design simulation. They are not covered by production test limits or characterization.

**Table 56-7.** BOD33 Characteristics on $V_{DD}$ and $V_{BAT}$ Monitoring in Low-Power Mode (During Standby/Backup/Hibernate Modes)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71 V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min.(1) | Typical | Max.(2) | Units | Conditions |
| REG7 | VBOD33 | VDDANA and VBAT BOD levels | 1.39 | — | 1.63 | V | ActiveLEVEL[7:0] = 0x00 (Min Value) , HYST[3:0] =0x02 |
| | | | 1.52 | — | 1.80 | V | ActiveLEVEL[7:0] = 0x19 (Recommended Value) , HYST[3:0] =0x02 |
| | | | 1.54 | — | 1.82 | V | ActiveLEVEL[7:0] = 0x1C (Fuse Value) , HYST[3:0] =0x02 |
| | | | 2.8 | — | 3.28 | V | ActiveLEVEL[7:0] = 0xFF(3) (Max Value) , HYST[3:0] = 0x00 |
| REG9 | $VBOD33_{LEVEL_STEP}$ | VBOD33 step size, LEVEL[7:0] | — | 6 | — | mV | |
| REG10 | $VBOD33_{HYST_STEP}$ | VBOD33 Hysteresis step size, HYST[3:0] | — | 6 | — | mV | |
| REG11 | TSTART(4) | Startup time (Time from enable to RDY) | — | 27 | — | µs | |

**Notes:**

1. BOD- = VBOD(min) = (1.5 + (LEVEL[7:0] * $VBOD33_{LEVEL_STEP}$).
2. BOD+ = VBOD(max) = (VBOD(min) + (HYST[3:0] * $VBOD33_{HYST_STEP}$)).
3. If a hysteresis value is programmed, the following equation must be respected to avoid any overflow on VBOD33(max): LEVEL[7:0] value < 255 - HYST[3:0] value.
4. These are based on design simulation. They are not covered by production test limits or characterization.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-8.** BOD33 Power Consumption

| Symbol | CPU Mode | Conditions | $T_A$ | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Active / Idle | VDD = 1.8V | Max 125°C Typ 25°C | 8.52 | 13.9 | µA |
| | | VDD = 3.3V | | 10.10 | 16.5 | |
| | Standby with BOD continuous normal mode | VDD = 1.8V | | 4.71 | 6.7 | |
| | | VDD = 3.3V | | 6.01 | 8.6 | |
| | Standby with BOD continuous low power mode or Hibernate mode | VDD = 1.8V | | 0.15 | 0.27 | |
| | | VDD = 3.3V | | 0.21 | 0.35 | |

## 56.5.3. Analog-to-Digital Converter (ADC) Characteristics (125°C)

**Table 56-9.** ADC Electrical Specifications (1)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **Device Supply** | | | | | | | |
| ADC1 | VDDANA | ADC Module Supply | VDDANA(min) | — | VDDANA(max) | V | VDDIOx = VDDANA |
| **Reference Inputs** | | | | | | | |
| ADC3 | VREF | ADC Reference Voltage | 1 | — | VDDANA-0.4 | V | Internal or External Reference VREF ≤ VDDANA - 0.4V |
| **Analog Input Range** | | | | | | | |
| ADC7 | AFS | Full-Scale Analog Input Signal Range (Single-Ended) | GNDANA | — | VREF | V | |
| ADC9 | | Full-Scale Analog Input Signal Range (Differential) | -VREF | — | +VREF | V | |
| ADC11 | VCM | Input common mode voltage | 0 | | VDDANA | V | CTRLA.R2R =1 |
| | | | See Note 2 | | | | CTRLA.R2R =0 |
| ADC13 | $T_{SETTING}$ | ADC stabilization Time | — | 10 | — | µs | CTRLA.ENABLE=1 or CTRLA.ONEDEMAND=1 |

**Notes:**

1. These values are based on simulation. They are not covered by production test limits or characterization.
2. Limit the input common mode voltage using the following equations (where, VCM_IN is the input channelcommon mode voltage):
   When CTRLA.R2R = 0:
   - VCM_IN < 0.75*VREF
   - VCM_IN > Maximum of (0, VREF-VDDANA-0.7, 1.25*VREF-VDDANA)

**Table 56-10.** ADC Conversion Timing Requirements (4)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **ADC Clock Requirements** | | | | | | | |
| ADC21 | TAD | ADC Clock Period | 62.5 | — | 3125 | ns | |
| ADC23 | fGCLK_ADCx | ADCx Module GCLK max input freq | — | — | 100 | MHz | |
| **ADC Single-Ended Throughput Rates** | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-10.** ADC Conversion Timing Requirements [4] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| ADC25 | FTP (Single-Ended Mode) | Throughput Rate [3] (Single-Ended) | 24.62 | — | 1231 | ksps | 12-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 26.67 | — | 1333 | | 10-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0[1] |
| | | | 32 | — | 1600 | | 8-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 20 | — | 1000 | ksps | 12-bit resolution, Rsource ≤6,550Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 21.30 | — | 1067 | | 10-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 24.62 | — | 1231 | | 8-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |
| **ADC Differential Mode Throughput Rates** | | | | | | | |
| ADC27 | FTP (Differential Mode) | Throughput Rate[3] (Differential Mode) | 24.62 | — | 1231 | ksps | 12-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 29.09 | — | 1455 | | 10-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 35.56 | — | 1778 | | 8-bit resolution, Rsource ≤147 Ω, SAMPCTRL.SAMPLEN=0 [1] |
| | | | 20 | — | 1000 | ksps | 12-bit resolution, Rsource ≤6,550Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 22.86 | — | 1143 | | 10-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |
| | | | 26.67 | — | 1333 | | 8-bit resolution, Rsource ≤6,550 Ω SAMPCTRL.SAMPLEN=n/a [2] |

**Notes:**

1. ADC_ Sample time = ((SAMPCTRL.SAMPLEN + 1) * TAD) and SAMPCTRL.OFFCOMP = 0.
2. ADC_ HDW forces sample time to 4*TAD when SAMPCTRL.OFFCOMP = 1, user SAMPCTRL.SAMPLEN is ignored.
3. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)). # of active analog channels used = 1.
4. These values are based on simulation and not covered by production test limits or characterization.

**Table 56-11.** ADC Sample Timing Requirements [4]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| ADC29 | TSAMP | ADC Sample Time [1,2,3] | 1[1] | — | — | TAD | TAD(min) Ext Analog Input Rsource ≤ 147 Ω |
| | | | 2[1] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 2,250 Ω |
| | | | 3[1] | — | — | | TAD(min), Ext Analog Input Rsource ≤ 4,400 Ω |
| | | | 4[1,2] | — | — | | TAD(min) , Ext Analog Input Rsource ≤ 6,550 Ω |
| | | | 5[1] | — | — | | TAD(min), Ext Analog Input Rsource ≤ 8,700 Ω |
| | | | 6[1] | — | — | | TAD(min), Ext Analog Input Rsource ≤ 10,850 Ω |
| | | | See Note 5 | | | ns | Sample time with DAC as input |
| | | | 10000 | — | — | | Sample time with Temp sensor or band gap as input |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-11.** ADC Sample Timing Requirements [4] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| ADC31 | TCNV | Conversion Time [3] (Single-Ended Mode) | | 12 | | TAD | 12-bit resolution |
| | | | | 11 | | | 10-bit resolution |
| | | | | 9 | | | 8-bit resolution |
| ADC33 | | Conversion Time [3] (Differential Mode) | | 12 | | TAD | 12-bit resolution |
| | | | | 10 | | | 10-bit resolution |
| | | | | 8 | | | 8-bit resolution |
| ADC35 | CSAMPLE | ADC Internal Sample Cap | 2 | — | 3 | pf | |
| ADC37 | RSAMPLE | ADC Internal impedance | — | — | 2000 | Ω | |

**Notes:**

1. When SAMPCTRL.OFFCOMP = 0:
   - TSAMP = (((RSAMPLE + RSOURCE) * CSAMPLE * 9.7) / TAD)+1 rounded down to nearest whole integer
   - User SAMPCTRL.SAMPLEN = (TSAMP - 1)
2. When SAMPCTRL.OFFCOMP=1:
   - TSAMP = 4 (Forced by HDW)
   - User SAMPCTRL.SAMPLEN = (n/a, Ignored by HDW)
3. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)).
4. These values are based on simulation and are not covered by production test limits or characterization.
5. See ts specified in DAC Electrical Characteristics.

**Table 56-12.** Differential Mode ADC Electrical Specifications [1]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature -40°C ≤ TA ≤ +125°C** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **DIFFERENTIAL MODE ADC Accuracy** | | | | | | | |
| DADC41 | Res | Resolution | 8 | — | 12 | bits | Selectable 8, 10, 12 bit Resolution Ranges |
| DADC43 | EN0B | Effective Number of bits | 10.5 | — | — | | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| DADC45 | | | 10.5 | — | — | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC47 | INL | Integral Nonlinearity | -1.8 | — | +1.8 | LSb | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| DADC49 | | | -1.9 | — | +1.9 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC51 | DNL | Differential Nonlinearity | -1.0 | — | +1.0 | | 1msps, R2R disabled,VREF = VDDANA = VDDIO = 3.0 |
| DADC53 | | | -1.0 | — | +1.2 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-12.** Differential Mode ADC Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| DADC55 | GERR | Gain Error with REFCTRL.REFCOMP=1 | - 0.21 | — | +0.20 | %FSR | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC57 | | | - 0.13 | — | +0.21 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC59 | | | - 10 | — | +6.7 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC61 | | | - 0.48 | — | +0.75 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC63 | | Gain Error with REFCTRL.REFCOMP=0 | - 0.30 | — | +0.20 | | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC65 | | | - 0.94 | — | +0.71 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC67 | | | - 10 | — | +6.7 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC69 | | | - 1.20 | — | +1.28 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC71 | E0FF | Offset Error with SAMPCTRL.OFFCOMP=1 | - 3.6 | — | +3.1 | mV | 1msps, Internal VREF = VDDANA = VDDIO = 3.0 |
| DADC73 | | | - 3.3 | — | +2.7 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC75 | | | - 3.6 | — | +2.9 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC77 | | | - 3.6 | — | +3.3 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC79 | | Offset Error with SAMPCTRL.OFFCOMP=0 | - 12.3 | — | +12.3 | | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| DADC81 | | | - 12.2 | — | +12.4 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| DADC83 | | | - 14.3 | — | +14.7 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| DADC85 | | | - 13.6 | — | +14.0 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| DADC87 | TUE[3] | Total Unadjusted Error | -5.2 | — | +5.2 | LSb | 1msps, R2R disabled, Internal VREF = VDDANA = VDDIO = 3.0 |
| DADC89 | | | -5.8 | — | +5.8 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| **DIFFERENTIAL MODE ADC Dynamic Performance** | | | | | | | |
| DADC91 | SINAD | Signal to Noise and Distortion | 65.3 | — | — | dB | VREF=VDDANA=VDDIO=3.0v @ 12bit max sampling rate, Fin = 14 kHz / Full range Input signal [2] |
| DADC93 | SNR | Signal to Noise ratio | 64.7 | — | — | | |
| DADC95 | SFDR | Spurious Free Dynamic Range | 76.6 | — | — | | |
| DADC97 | THD | Total Harmonic Distortion | — | — | -78.6 | | |
| DADC99 | Nrms | Noise RMS | — | — | 2.4 | mV | External VREF = 2.0V, VDDANA = VDDIO = 3.0, Constant Input Voltage |
| | | | | | 2.5 | | VREF = VDDANA = VDDIO = 3.0, Constant Input Voltage |

**Notes:**

1. These values are based on characterization. These values are not covered by test limits in production.
2. All values expressed in decibel refer to the full scale input and are tested with an input signal 0.35dB below full scale; THD measured on the first seven harmonics of the input signal.
3. With REFCTRL.REFCOMP=1 and SAMPCTRL.OFFCOMP=1.

**Table 56-13.** Single Ended Mode ADC Electrical Specifications [1]

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| **SINGLE ENDED MODE ADC Accuracy** | | | | | | | |
| SADC41 | Res | Resolution | 8 | — | 12 | bits | Selectable 8, 10, 12 bit Resolution Ranges |
| SADC43 | EN0B | Effective Number of bits | 8.9 | — | — | | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| SADC45 | | | 8.9 | — | — | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-13.** Single Ended Mode ADC Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=VDDANA 1.71V to 3.63V (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +125°C | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typical | Max. | Units | Conditions |
| SADC47 | INL | Integral Nonlinearity | -3.2 | — | +3.2 | LSb | 1msps, R2R disabled, VREF = VDDANA = VDDIO = 3.0 |
| SADC49 | | | -3.9 | — | +3.9 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC51 | DNL | Differential Nonlinearity | -1.0 | — | +1.0 | | 1msps, R2R disabled,VREF = VDDANA = VDDIO = 3.0 |
| SADC53 | | | -1.0 | — | +1.5 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC55 | GERR | Gain Error with REFCTRL.REFCOMP=1 | - 0.3 | — | +0.3 | %FSR | 1msps, VREF = VDDANA = VDDIO = 3.0 |
| SADC57 | | | - 0.16 | — | +0.3 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC59 | | | - 11 | — | +7 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| SADC61 | | | - 0.5 | — | +0.7 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| SADC71 | E0FF | Offset Error with SAMPCTRL.OFFCOMP=1 | - 21 | — | +9.3 | mV | 1msps, Internal VREF = VDDANA = VDDIO = 3.0 |
| SADC73 | | | - 21 | — | +17 | | 1msps, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| SADC75 | | | - 25 | — | +26 | | 1msps, VREF=1V Internal band gap, VDDANA = VDDIO = 3.0 |
| SADC77 | | | - 27 | — | +24 | | 1msps, VREF = VDDANA/2, VDDANA = VDDIO = 3.0 |
| SADC87 | TUE[3] | Total Unadjusted Error | -21 | — | +21 | LSb | 1msps, R2R disabled, Internal VREF = VDDANA = VDDIO = 3.0 |
| SADC89 | | | -19.7 | — | +19.7 | | 1msps, R2R disabled, External VREF = 2.0V, VDDANA = VDDIO = 3.0 |
| **SINGLE ENDED MODE ADC Dynamic Performance** | | | | | | | |
| SADC91 | SINAD | Signal to Noise and Distortion | 55.6 | — | — | dB | VREF=VDDANA=VDDIO=3.0v @ 12bit max sampling rate, Fin = 14 kHz / Full range Input signal [2] |
| SADC93 | SNR | Signal to Noise ratio | 54.7 | — | — | | |
| SADC95 | SFDR | Spurious Free Dynamic Range | 67.9 | — | — | | |
| SADC97 | THD | Total Harmonic Distortion | — | — | -65.8 | | |
| SADC99 | Nrms | Noise RMS | — | — | 2.3 | mV | External VREF = 2.0V, VDDANA = VDDIO = 3.0, Constant Input Voltage |
| | | | | | 2.45 | | VREF = VDDANA = VDDIO = 3.0, Constant Input Voltage |

**Notes:**

1. These values are based on characterization. These values are not covered by test limits in production.
2. All values expressed in decibel refer to the full scale input and are tested with an input signal 0.35dB below full scale; THD measured on the first seven harmonics of the input signal.
3. With REFCTRL.REFCOMP=1 and SAMPCTRL.OFFCOMP=1.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-14.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ $V_{DDANA}$ | Differential mode | fs = 1 Msps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | Max 125°C Typ 25°C | 279 | 848 | µA |
| | | fs = 1 Msps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 482 | 1381 | |
| | | fs = 10 ksps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 28 | 91 | |
| | | fs = 10 ksps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 241 | 807 | |
| | Single Ended mode | fs = 1 Msps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | Max 125°C Typ 25°C | 307 | 820 | µA |
| | | fs = 1 Msps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 499 | 1156 | |
| | | fs = 10 ksps / Reference buffer disabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 38 | 108 | |
| | | fs = 10 ksps / Reference buffer enabled / BIASREFBUF = '111', BIASREFCOMP = '111' $V_{DDANA}$ = $V_{REF}$ = 3.0V | | 245 | 881 | |

## 56.5.4. Digital-to-Analog Converter (DAC) Characteristics (125°C)

**Table 56-15.** Differential Mode (1)

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| INL | Integral Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V, External Ref. = 2.0V, $C_{LOAD}$ = 50 pF | - | ±2.4 | ±4.1 | LSB |
| | | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V, Internal Ref , $C_{LOAD}$ = 50 pF | - | ±3.2 | ±4.2 | |
| DNL | Differential Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V,External Ref. = 2.0V, $C_{LOAD}$ = 50 pF | - | ±2.4 | ±4.5 | LSB |
| | | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V, Internal Ref , $C_{LOAD}$ = 50 pF | - | ±3.5 | ±5.4 | |
| Gerr | Gain Error | External Reference voltage | - | ±0.4 | ±1.9 | % FSR |
| | | 1.0V Internal Reference voltage | - | ±0.8 | ±8.5 | |
| Offerr | Offset Error | External Reference voltage | - | ±13 | ±47 | mV |
| | | 1.0V Internal Reference voltage | - | ±8 | ±79 | |
| ENOB | Effective Number of Bits | Fs = 1Ms/s - External Ref - CCTRL=0x2 | 9.9 | 10.7 | 10.9 | dB |
| SNR | Signal to Noise ratio | | 63.5 | 68.6 | 72.6 | |
| THD | Total Harmonic Distortion | | -79.1 | -72.5 | -61.0 | |

**Note:**

1. These values are based on characterization. These values are not covered by test limits in production.

**Table 56-16.** Single-Ended Mode (1)

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| INL | Integral Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V External Ref. = 2.0V, $C_{LOAD}$ = 50 pF | - | ±2.7 | ±6.0 | LSB |
| | | i12clk = 12 MHz $V_{DDANA}$ = 3.0V, Internal Ref , $C_{LOAD}$ = 50 pF | - | ±5.2 | ±11.2 | |
| DNL | Differential Non Linearity, Best-fit curve from 0x080 to 0xF7F | i12clk = 12 MHz, $V_{DDANA}$ = 3.0V External Ref = 2.0V, $C_{LOAD}$ = 50 pF | - | ±3.5 | ±8.1 | LSB |
| | | i12clk = 12 MHz $V_{DDANA}$ = 3.0V, Internal Ref, $C_{LOAD}$ = 50 pF | - | ±6.4 | ±12.1 | |
| Gerr | Gain Error | External Reference voltage | - | ±0.3 | ±1.6 | % FSR |
| | | 1.0V Internal Reference voltage | - | ±0.8 | ±8.6 | |
| Offerr | Offset Error | External Reference voltage | - | ±7 | ±25.5 | mV |
| | | 1.0V Internal Reference voltage | - | ±2 | ±19 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Table 56-16. Single-Ended Mode (1) (continued) | | | | | | |
|---|---|---|---|---|---|---|
| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
| ENOB | Effective Number of Bits | Fs = 1Ms/s - External Ref - CCTRL=0x2 | 9.1 | 10.3 | 10.7 | dB |
| SNR | Signal to Noise ratio | | 63.5 | 68.6 | 72.6 | |
| THD | Total Harmonic Distortion | | -79.1 | -72.8 | -61.0 | |

**Note:**

1. These values are based on characterization. These values are not covered by test limits in production.

**Table 56-17.** Power Consumption

| Symbol | Parameters | Conditions | TA | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|---|
| $I_{DDANA}$ | Differential Mode, DC supply current, 2 output channels - without load | fs = 1 Msps, CCTR L= 0x2, $V_{REF}$ > 2.4V, $V_{DD}$= 3.3V | Max. 125°C Typ. 25°C | - | 384 | 634 | µA |
| | | fs = 10 ksps, CCTRL = 0x0, $V_{REF}$ < 2.4V, $V_{DD}$= 3.3V | | - | 283 | 482 | |
| | Single-Ended Mode, DC supply current, 2 output channels - without load | fs = 1 Msps, CCTRL = 0x2, $V_{REF}$ > 2.4V, $V_{DD}$ = 3.3V | | - | 306 | 517 | µA |
| | | fs = 10 ksps, CCTRL = 0x0, $V_{REF}$ < 2.4V, $V_{DD}$ = 3.3V | | - | 230 | 389 | |

## 56.5.5. Analog Comparator (AC) Characteristics (125°C)

**Table 56-18.** Analog Comparator Characteristics

| Symbol | Parameters | Conditions | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| PNIVR | Positive and Negative input range voltage | - | 0 | - | VDDANA | V |
| ICMR | Input common mode range | - | 0 | - | VDDANA-0.2 | V |
| Off (1) | Offset | High-speed COMPCTRLn.SPEED = 0x3 | -22 | +/-3 | 22 | mV |
| Vhys (2) | Hysteresis | High-speed COMPCTRLn.SPEED = 0x3 COMPCTRLn.HYST = 0x0 (HYST25) | 12 | 25 | 47 | mV |
| | | High-speed COMPCTRLn.SPEED = 0x3 COMPCTRLn.HYST = 0x1 (HYST50) | 28 | 50 | 80 | |
| | | High-speed COMPCTRLn.SPEED = 0x3 COMPCTRLn.HYST = 0x2 (HYST75) | 36 | 75 | 114 | |
| | | High-speed COMPCTRLn.SPEED = 0x3 COMPCTRLn.HYST = 0x3 (HYST100) | 61 | 100 | 148 | |
| Tpd (3) | Propagation Delay Vcm = Vddana/2, Vin = +-100mV overdrive from Vcm | High-speed COMPCTRLn.SPEED = 0x3 | - | 24.1 | 42 | ns |
| Tstart | Startup time | High-speed COMPCTRLn.SPEED = 0x3 | - | 4.7 | 8 | µs |

**Notes:**

1. Hysteresis is disabled.
2. Hysteresis is enabled.
3. Tpd is measured from Vin transition to ACOUT (AC direct output) toggle.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 56-19.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $I_{DDANA}$ | Current consumption for One AC enabled, Hysteresis disabled voltage scaler disabled | COMPCTRLn.SPEED = 0x3, $V_{DDANA}$ = 3.3V | Max.125°C Typ.25°C | 59 | 106 | µA |
| | Current consumption Voltage Scaler only | $V_{DDANA}$=3.3V | | 11 | 23.3 | |

### 56.5.6. PTC Characteristics

The values in the following Power Consumption table are measured values of power consumption under the following conditions:

**Operating Conditions:**

VDD = 3.0V

**Clocks**

DFLL48M used as main clock source, running undivided at 48 MHz

CPU is running on Flash with 2 wait states, at 48 MHz

PTC running at 4 MHz

**PTC Configuration**

Mutual Capacitance mode

One touch channel

**System Configuration**

Standby Sleep mode enabled

RTC running on ULP32K: used to define the PTC scan rate, through the event system

RTC interrupts (wake up) the CPU to perform PTC scans

**Table 56-20.** Power Consumption (1)

| Symbol | Parameters | PTC scan rate (msec) | Oversamples | $T_A$ | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| IDD | Current Consumption | 10 | 4 | Max. 125°C Typ 25°C | 137 | 3960 | µA |
| | | | 16 | | 146 | 3989 | |
| | | 50 | 4 | | 77 | 3882 | |
| | | | 16 | | 79 | 3893 | |
| | | 100 | 4 | | 68 | 3877 | |
| | | | 16 | | 69 | 3885 | |
| | | 200 | 4 | | 64 | 3870 | |
| | | | 16 | | 65 | 3872 | |

**Note:**

1. These values are based on characterization.

## 56.6. NVM Characteristics (125°C)

**Table 56-21.** NVM Flash Read Wait States for Worst Case Conditions

| Wait States (WS) | 0 WS | 1 WS | 2 WS | 3 WS | 4 WS | 5 WS | 6 WS | Auto WS |
|---|---|---|---|---|---|---|---|---|
| Read Operations | 1 cycle | 2 cycles | 3 cycles | 4 cycles | 5 cycles | 6 cycles | 7 cycles | n cycles |
| CPU FMax (MHz)(1) | 19 | 38 | 57 | 76 | 95 | 100 | 100 | 100 |

**Note:**

1. VDD > 1.71.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Maximum operating frequencies are given in the table above, but are limited by the Embedded Flash access time when the processor is fetching code out of it. Theses tables provide the device maximum operating frequency defined by the field RWS of the NVMCTRL CTRLA register when automatic wait states (AUTOWS) is disabled. This field defines the number of Wait states required to access the Embedded Flash Memory.

## 56.7. Oscillators Characteristics (125°C)

### 56.7.1. Crystal Oscillator (XOSC) Characteristics (125°C)

**Table 56-22.** Multi Crystal Oscillator Electrical Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| Tstart | Startup time | F = 8MHz - CL=20 pF - Cshunt = 2 pF - IMULT=0x3 | - | 39700 | 72200 | Cycles |
| | | F = 16MHz - CL=20 pF - Cshunt = 1,5 pF - IMULT=0x4 | - | 37550 | 73000 | |
| | | F = 24MHz - CL=20 pF - Cshunt = 2,5 pF - IMULT=0x5 | - | 32700 | 71000 | |
| | | F = 48MHz - CL=13 pF - Cshunt = 5 pF - IMULT=0x6 | - | 18400 | 38500 | |

**Table 56-23.** Power Consumption

| Symbol | Parameters | Conditions | TA | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | F = 8 MHz - $C_L$ = 20 pF - IMULT = 0x3, ENALC = OFF | Max. 125°C, Typ. 25°C | 0.43 | 2.27 | mA |
| | | ENALC = ON | | 0.16 | 1.87 | |
| | | F = 16 MHz - $C_L$ = 20 pF - IMULT = 0x5, ENALC = OFF | | 1.31 | 3.72 | |
| | | ENALC = ON | | 0.25 | 2.23 | |
| | | F = 32 MHz - $C_L$ = 13 pF - IMULT = 0x5, ENALC = OFF | | 2.92 | 6.49 | |
| | | ENALC = ON | | 0.40 | 2.43 | |
| | | F = 48 MHz - $C_L$ = 13 pF - IMULT = 0x6, ENALC = OFF | | 2.70 | 6.71 | |
| | | ENALC = ON | | 0.76 | 3.52 | |

### 56.7.2. External 32 kHz Crystal Oscillator (XOSC32K) Characteristics (125°C)

**Table 56-24.** Power Consumption

| Symbol | Parameter | Conditions | TA | Gain Mode | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current consumption | VDD=3.0V | Max 125°C Typ 25°C | Std. | 1.5 | 2.6 | µA |
| | | | | High | 1.9 | 3.4 | |

### 56.7.3. Internal Ultra-Low Power 32 kHz RC Oscillator (OSCULP32K) Characteristics (125°C)

**Table 56-25.** Ultra-Low Power Internal 32 kHz RC Oscillator Electrical Characteristics

| Symbol | Parameter | Conditions | Min. | Typ. | Max | Units |
|---|---|---|---|---|---|---|
| $F_{OUT}$ | Output frequency | [-40, +125]°C, VDDANA>1.71V | 26.00 | 32.768 | 40.92 | kHz |

### 56.7.4. Digital Frequency Locked Loop (DFLL48M) Characteristics (125°C)

**Table 56-26.** DFLL48M Characteristics - Open Loop Mode (1)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $F_{OpenOUT}$ | Output frequency | DFLLVAL after Reset<br>LDO Regulator mode, [-40, 125]°C | 45.57 | 48 | 50.63 | MHz |
| | | DFLLVAL after Reset<br>LDO Regulator mode, [0, 60]°C | 47.12 | 48 | 48.9 | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:**

1. DFLL48 in open loop can be used only with LDO regulator.

**Table 56-27.** DFLL48M Power Consumption

| Symbol | Parameter | Conditions | TA | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | Open Loop mode - DFLLVAL after reset VDD = 3.3V | Max. 125°C Typ. 25°C | - | 400 | 2129 | µA |
| | | Closed Loop mode - $f_{REF}$ = 32 .768 kHz VDD = 3.3V | | - | 404 | 2113 | µA |

### 56.7.5. Fractional Digital Phase Lock Loop (FDPLL) Characteristics (125°C)

**Table 56-28.** Fractional Digital Phase Lock Loop Characteristics (1)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| Jp | Period jitter (Peak-Peak value) | $f_{IN}$ = 32 kHz, $f_{OUT}$ = 96 MHz | - | 1.9 | 3.0 | % |
| | | $f_{IN}$ = 32 kHz, $f_{OUT}$ = 200 MHz | - | 3.4 | 6.0 | |
| | | $f_{IN}$ = 3.2 MHz, $f_{OUT}$ = 96 MHz | - | 2.0 | 3.1 | |
| | | $f_{IN}$ = 3.2 MHz, $f_{OUT}$ = 200 MHz | - | 4.3 | 7.2 | |

**Note:**

1. These FDPLL200Mn characteristics are applicable with LDO regulator and a direct reference (i.e., REFCLK is XOSC or XOSC32K, not GCLK).

**Table 56-29.** Fractional Digital Phase Lock Loop Power Consumption

| Symbol | Parameter | Conditions | TA | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $I_{DD}$ | Current Consumption | Ck = 96 MHz, VDD = 3.3V | Max. 125°C Typ. 25°C | 0.9 | 2.5 | mA |
| | | Ck = 200 MHz, VDD = 3.3V | | 2.0 | 3.4 | |

## 56.8. Timing Characteristics (125°C)

### 56.8.1. SERCOM in SPI Mode Timing (125°C)

**Table 56-30.** SPI Timing Characteristics and Requirements(1)

| Symbol | Parameter | Conditions | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|---|
| $t_{MIS}$ | MISO setup to SCK | Host, VDD>2.70V | 19.5 | - | - | ns |
| | | Host, VDD>1.71V | 20 | - | - | |
| $t_{SOV}$ | MISO output valid SCK | Client, VDD>2.70V | 16.5 | - | - | ns |
| | | Client, VDD>1.71V | 25 | - | - | |

1. These values are based on simulation, with capacitance load between 5pF and 20pF. These values are not covered by test limits in production.


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 57. AEC Q-100 Grade 1, 125°C Electrical Characteristics

**Important:** AEC-Q100 Grade 1 Electrical Specifications are covered by Electrical Characteristics at 125°C unless explicitly mentioned in this document.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

# 58. Appendix A

## 58.1. ISELED FULL License Enabled Functional Devices

Microchip offers ISELED full license enabled devices which can utilize the ISELED software stack and library available for download from the Microchip Website. Included in the download will be the ISELED stack source code file and header, the library binary, and the library user's manual. Refer to the Inova data sheet "INLC10AQ (ISELED control commands; document AN-INLC_04)". For additional information on Microchip's ISELED solutions, please visit the Microchip Website (www.microchip.com/iseled).

Contact Microchip Sales Office for more details on the ISELED full license enabled devices or to request a part number which is not available in the section below "Ordering Information".

## 58.2. Ordering Information

The following table lists the ISELED full license enabled devices which can utilize the ISELED full license software stack and library.

**Table 58-1.** SAM E51N

| Ordering Code | FLASH (bytes) | SRAM (bytes) | Temperature Grade | Package | Carrier Type |
|---|---|---|---|---|---|
| ATSAME51N19A-AZT510 | 512K | 192K | -40°C to 125°C | TQFP100 | Tape & Real |
| ATSAME51N19A-AZ510 | 512K | 192K | -40°C to 125°C | TQFP100 | Tray |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 59. Packaging Information

## 59.1. Package Marking Information

All devices are marked with Atmel logo and ordering code.

Additional marking information is as follows:

- "YY": Manufacturing year
- "WW": Manufacturing week
- "R": Internal Code
- "XXXXXX": Lot number




## 59.2. Thermal Considerations

### 59.2.1. Thermal Resistance Data

The following table summarizes the thermal resistance data depending on the package.

**Table 59-1.** Thermal Resistance Data

| Package Type | $\theta_{JA}$ | $\theta_{JC}$ |
|---|---|---|
| 64-pin TQFP | 57.4°C/W | 10.6°C/W |
| 100-pin TQFP | 55.0°C/W | 11.1°C/W |
| 128-pin TQFP | 48.7°C/W | 9.4°C/W |
| 120-pin TFBGA | 36.63°C/W | 12.2°C/W |
| 48-pin VQFN | 29.8°C/W | 10.0°C/W |
| 64-pin VQFN | 30.3°C/W | 9.9°C/W |
| 64-pin WLCSP | 36.8°C/W | 5.0°C/W |

### 59.2.2. Junction Temperature

**Table 59-2.** Thermal Operation Conditions

| Rating | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| **Industrial Temperature Devices** | | | | | |
| Operating Ambient Temperature Range | $T_A$ | -40 | — | 85 | °C |
| Operating Junction Temperature Range | $T_J$ | — | — | 105 | °C |
| **V-temp Temperature Devices** | | | | | |
| Operating Ambient Temperature Range | $T_A$ | -40 | — | 105 | °C |
| Operating Junction Temperature Range | $T_J$ | — | — | 125 | °C |
| **Extended Temperature Range** | | | | | |
| Operating Ambient Temperature Range | $T_A$ | -40 | — | 125 | °C |
| Operating Junction Temperature Range | $T_J$ | — | — | 145 | °C |
| **Power Dissipation:** | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 59-2.** Thermal Operation Conditions (continued)

| Rating | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Internal Chip Power Dissipation:<br>PINT = VDD x (IDD –∑ IOH)<br>I/O Pin Power Dissipation:<br>PI/O = ∑ (({VDD – VOH} x IOH) + ∑ (VOL x IOL)) | $P_D$ | | PINT + PI/O | | W |
| Maximum Allowed Power Dissipation | PDMAX | | (TJ – TA)/θJA | | W |

**Notes:**

1. $\theta_{JA}$ = Package thermal resistance, Junction-to-ambient (°C/W), see Thermal Resistance Data.
2. When heatsink is used, PDMAX = (TJ-TA)/($\theta_{HEATSINK}$+$\theta_{JC}$). Here $\theta_{HEATSINK}$ = Thermal resistance (°C/W) specification of the external cooling device. $\theta_{JC}$= Package thermal resistance, Junction-to-case thermal resistance (°C/W). See Table Thermal Resistance Data for additional information.

## 59.3. Package Drawings

**Note:** For current package drawings, refer to the Microchip Packaging Specification, which is available at http://www.microchip.com/packaging.

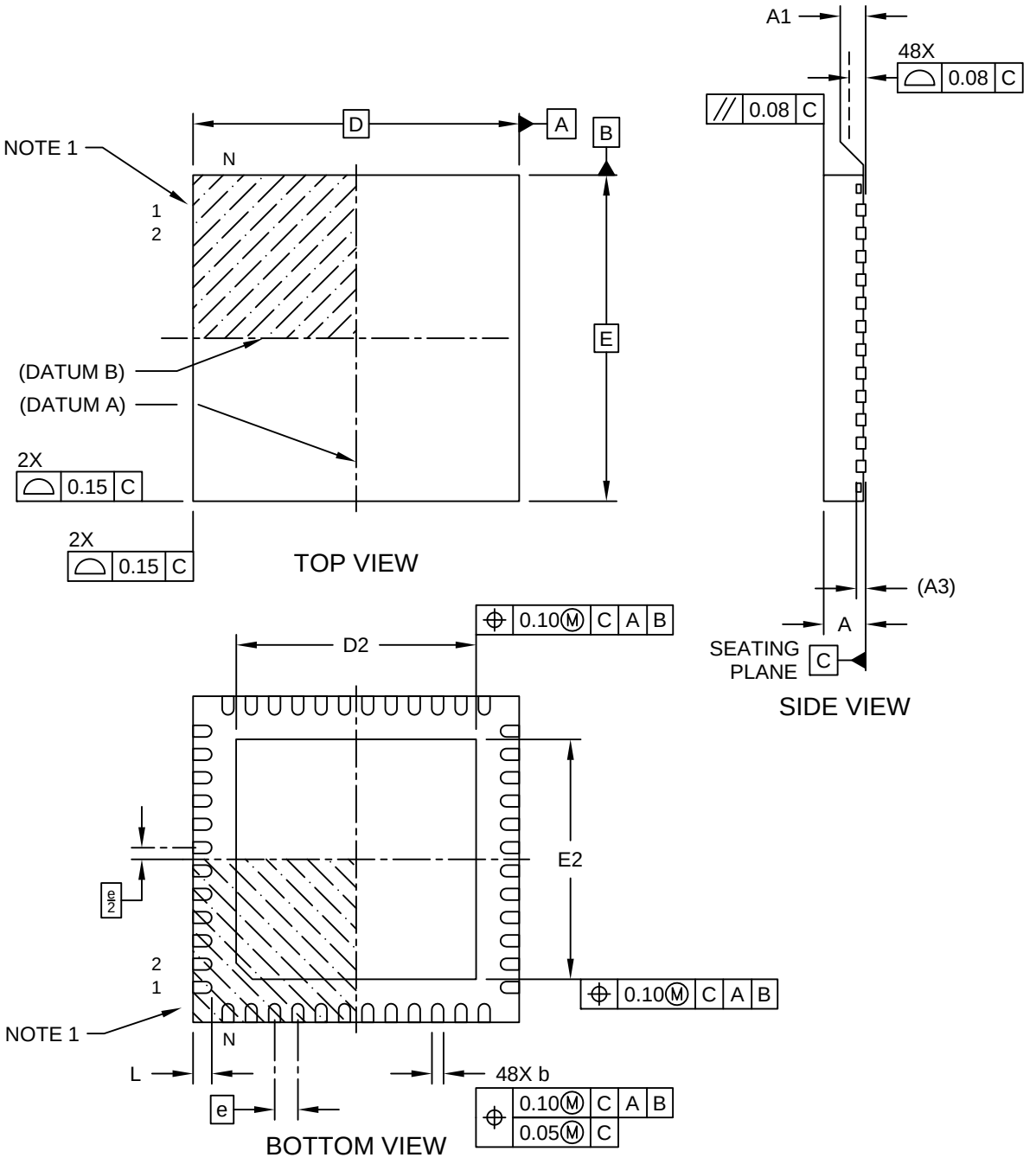

© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 59.3.1. 48-Pin VQFN

**48-Lead Very Thin Quad Flat Pack No-Leads (T3B) 7x7x0.9 mm [VQFN]**
**With 5.15x5.15 mm Exposed Pad; Atmel Legacy Global Package Code ZLG**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|

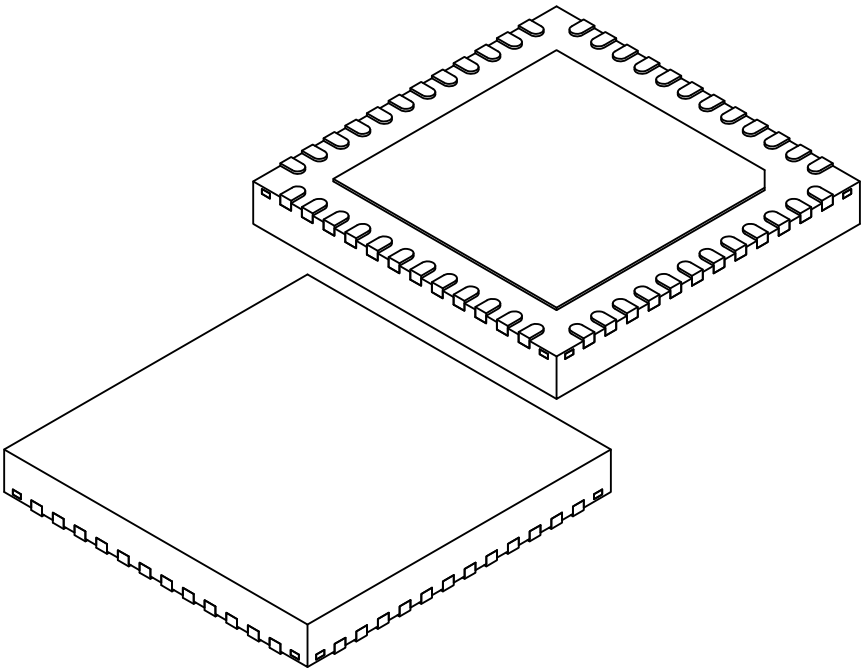



Microchip Technology Drawing C04-21425 Rev A Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48-Lead Very Thin Quad Flat Pack No-Leads (T3B) 7x7x0.9 mm [VQFN] With 5.15x5.15 mm Exposed Pad; Atmel Legacy Global Package Code ZLG

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Terminals | N | | 48 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | 0.80 | 0.85 | 0.90 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Terminal Thickness | A3 | | 0.20 REF | |
| Overall Length | D | | 7.00 BSC | |
| Exposed Pad Length | D2 | 5.05 | 5.15 | 5.25 |
| Overall Width | E | | 7.00 BSC | |
| Exposed Pad Width | E2 | 5.05 | 5.15 | 5.25 |
| Terminal Width | b | 0.18 | 0.25 | 0.30 |
| Terminal Length | L | 0.30 | 0.40 | 0.50 |
| Terminal-to-Exposed-Pad | K | 0.20 | - | - |

Notes:

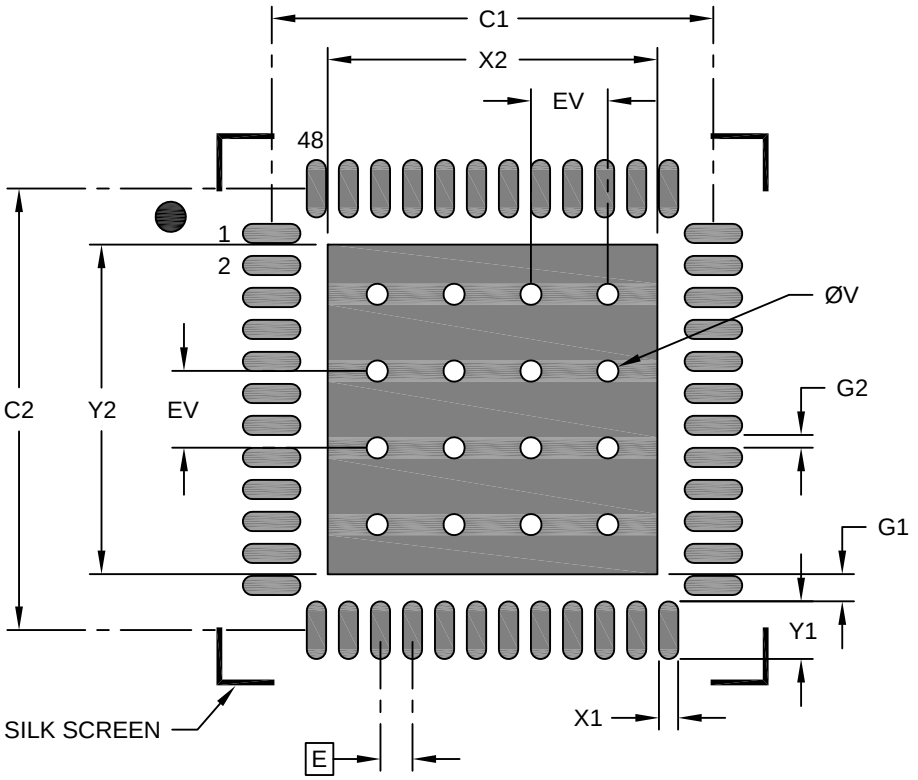
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21425 Rev A Sheet 2 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48-Lead Very Thin Quad Flat Pack No-Leads (T3B) 7x7x0.9 mm [VQFN] With 5.15x5.15 mm Exposed Pad; Atmel Legacy Global Package Code ZLG

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Optional Center Pad Width | X2 | | | 5.15 |
| Optional Center Pad Length | Y2 | | | 5.15 |
| Contact Pad Spacing | C1 | | 6.90 | |
| Contact Pad Spacing | C2 | | 6.90 | |
| Contact Pad Width (X20) | X1 | | | 0.30 |
| Contact Pad Length (X20) | Y1 | | | 0.90 |
| Contact Pad to Center Pad (X48) | G1 | 0.20 | | |
| Contact Pad to Contact Pad (X44) | G2 | 0.20 | | |
| Thermal Via Diameter | V | | 0.33 | |
| Thermal Via Pitch | EV | | 1.20 | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23425 Rev A

**Note:** The exposed die attach pad is not connected electrically inside the device. It is recommended that the exposed pad be connected to ground in the PCB.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 59-3.** Device and Package Maximum Weight

| | |
|---|---|
| 140 | mg |

**Table 59-4.** Package Reference

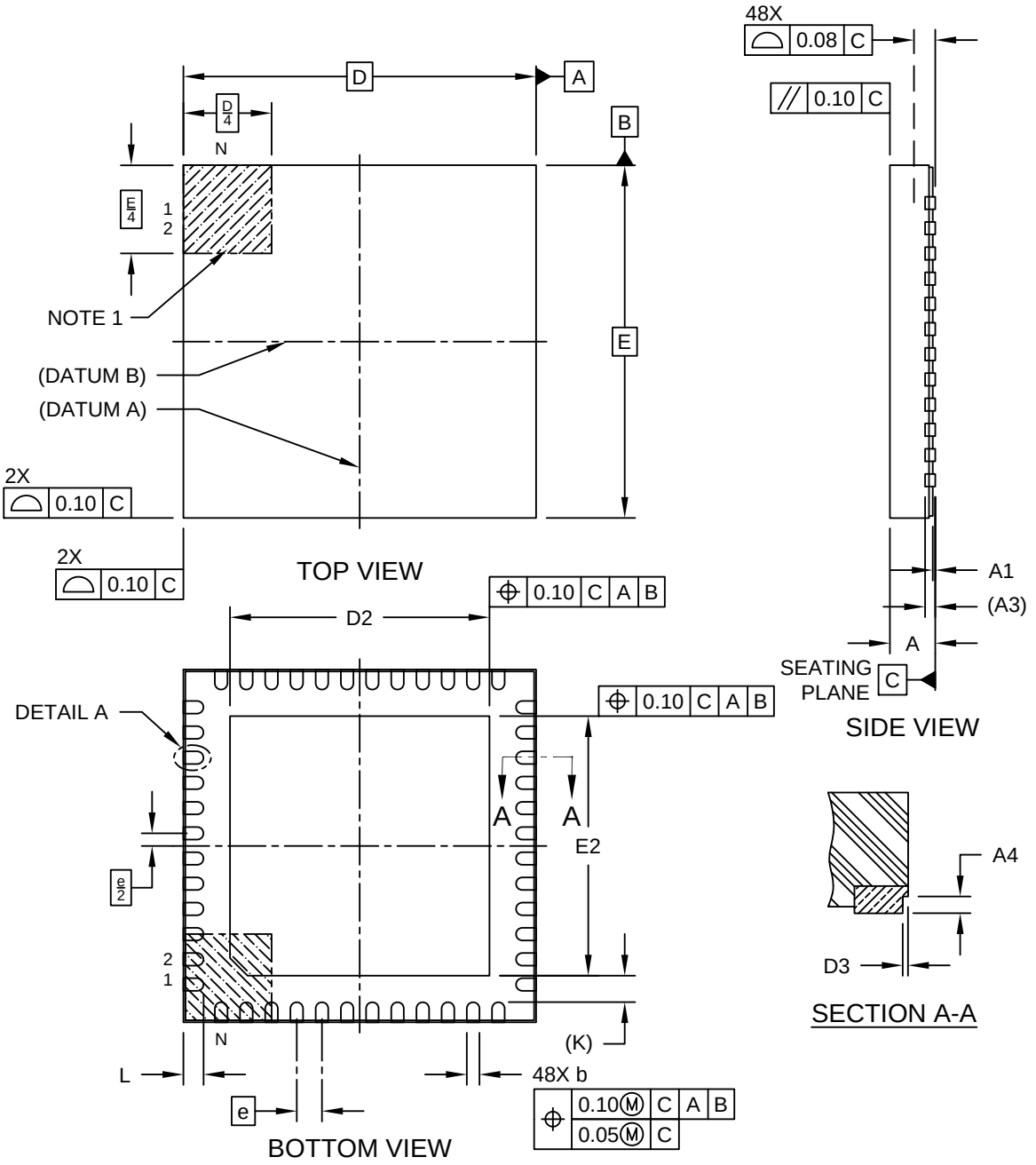
| | |
|---|---|
| Package Outline Drawing MCHP reference | C04-21425 |
| JESD97 Classification | E3 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 59.3.2. 48-Pin VQFN Wettable Flanks

**48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|

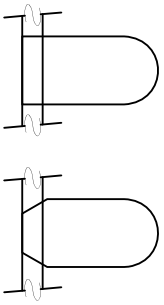



Microchip Technology Drawing C04-21493 Rev A Sheet 1 of 2

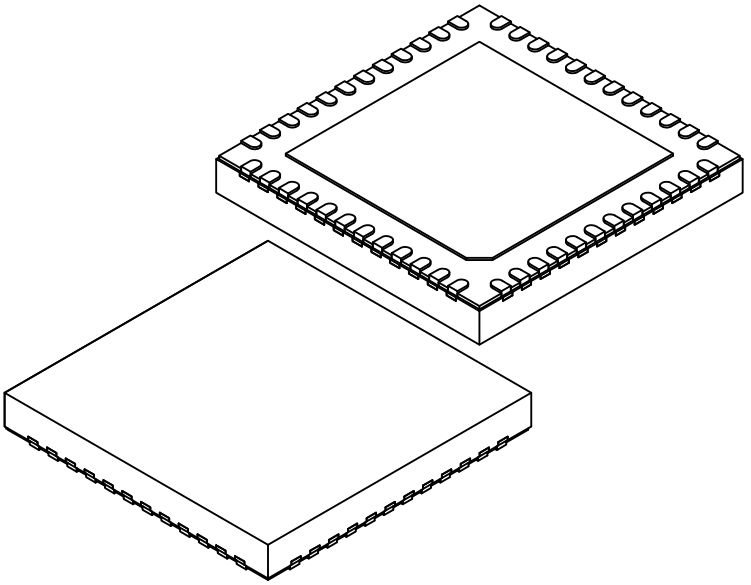

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



DETAIL 1
ALTERNATE TERMINAL CONFIGURATIONS



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Terminals | N | | 48 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | 0,80 | 0.85 | 0.90 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Terminal Thickness | A3 | | 0.203 REF | |
| Overall Length | D | | 7.00 BSC | |
| Exposed Pad Length | D2 | 5.05 | 5.15 | 5.25 |
| Overall Width | E | | 7.00 BSC | |
| Exposed Pad Width | E2 | 5.05 | 5.15 | 5.25 |
| Terminal Width | b | 0.20 | 0.25 | 0.30 |
| Terminal Length | L | 0.35 | 0.40 | 0.45 |
| Terminal-to-Exposed-Pad | K | | 0.53 REF | |
| Wettable Flank Step Length | D3 | - | - | 0.085 |
| Wettable Flank Step Height | A4 | 0.10 | - | 0.19 |

Notes:

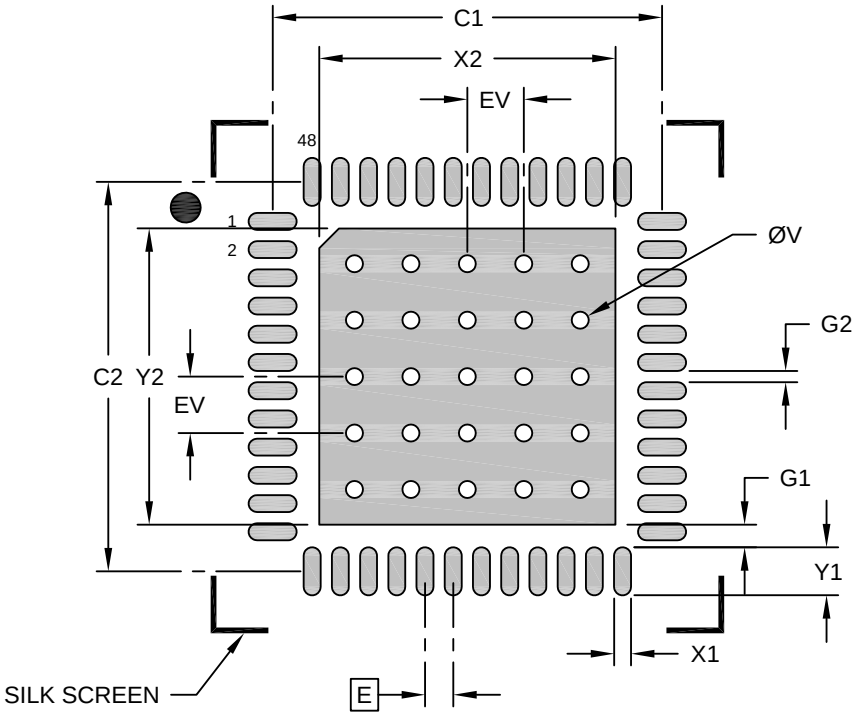
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21493 Rev A Sheet 2 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48-Lead Very Thin Plastic Quad Flat, No Lead Package (U5B) - 7x7 mm Body [VQFN] With 5.15 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZLH

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Optional Center Pad Width | X2 | | | 5.25 |
| Optional Center Pad Length | Y2 | | | 5.25 |
| Contact Pad Spacing | C1 | | 6.90 | |
| Contact Pad Spacing | C2 | | 6.90 | |
| Contact Pad Width (X48) | X1 | | | 0.30 |
| Contact Pad Length (X48) | Y1 | | | 0.85 |
| Contact Pad to Center Pad (X48) | G1 | 0.20 | | |
| Contact Pad to Center Pad (X44) | G2 | 0.40 | | |
| Thermal Via Diameter | V | | 0.30 | |
| Thermal Via Pitch | EV | | 1.00 | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

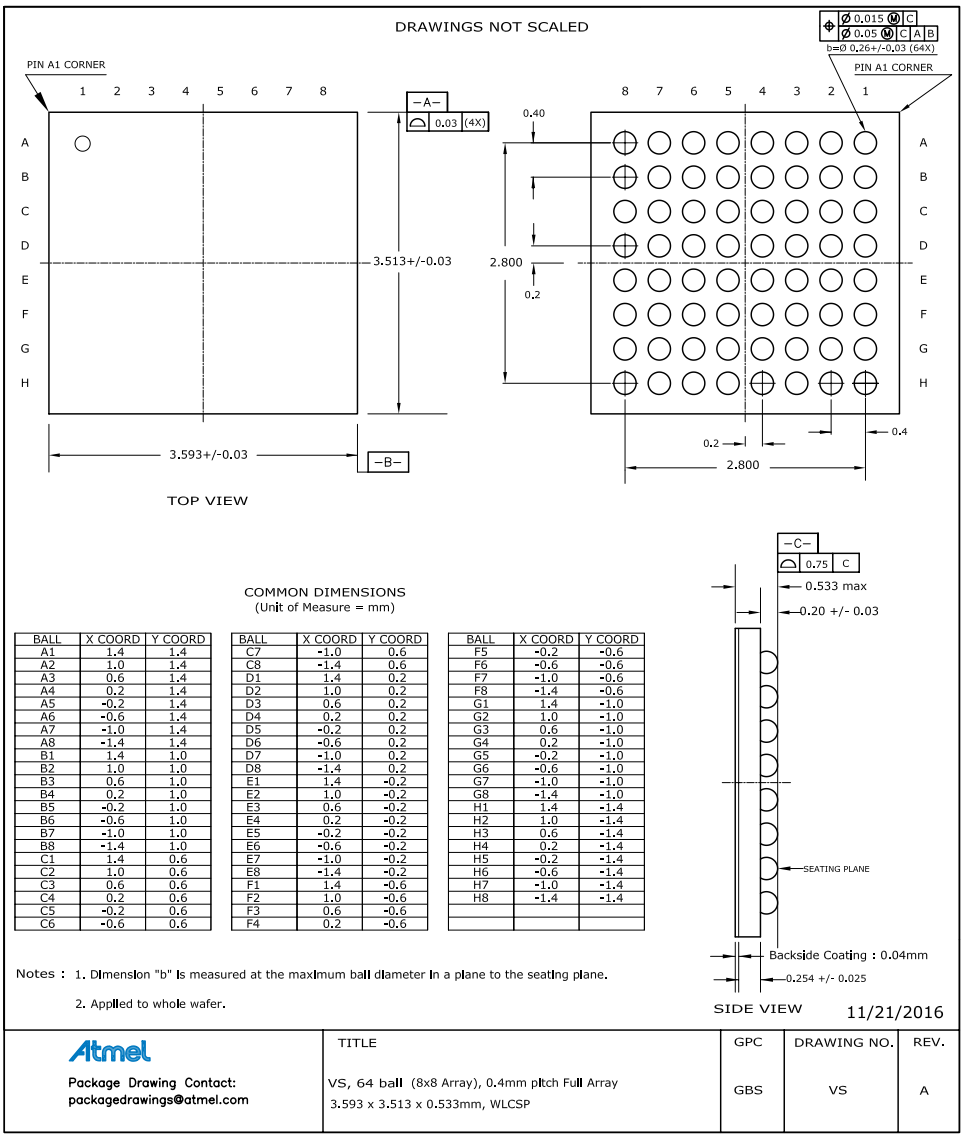
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-23493 Rev A


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 59.3.3. 64-Ball WLCSP




COMMON DIMENSIONS
(Unit of Measure = mm)

| BALL | X COORD | Y COORD |
|---|---|---|
| A1 | 1.4 | 1.4 |
| A2 | 1.0 | 1.4 |
| A3 | 0.6 | 1.4 |
| A4 | 0.2 | 1.4 |
| A5 | -0.2 | 1.4 |
| A6 | -0.6 | 1.4 |
| A7 | -1.0 | 1.4 |
| A8 | -1.4 | 1.4 |
| B1 | 1.4 | 1.0 |
| B2 | 1.0 | 1.0 |
| B3 | 0.6 | 1.0 |
| B4 | 0.2 | 1.0 |
| B5 | -0.2 | 1.0 |
| B6 | -0.6 | 1.0 |
| B7 | -1.0 | 1.0 |
| B8 | -1.4 | 1.0 |
| C1 | 1.4 | 0.6 |
| C2 | 1.0 | 0.6 |
| C3 | 0.6 | 0.6 |
| C4 | 0.2 | 0.6 |
| C5 | -0.2 | 0.6 |
| C6 | -0.6 | 0.6 |
| C7 | -1.0 | 0.6 |
| C8 | -1.4 | 0.6 |
| D1 | 1.4 | 0.2 |
| D2 | 1.0 | 0.2 |
| D3 | 0.6 | 0.2 |
| D4 | 0.2 | 0.2 |
| D5 | -0.2 | 0.2 |
| D6 | -0.6 | 0.2 |
| D7 | -1.0 | 0.2 |
| D8 | -1.4 | 0.2 |
| E1 | 1.4 | -0.2 |
| E2 | 1.0 | -0.2 |
| E3 | 0.6 | -0.2 |
| E4 | 0.2 | -0.2 |
| E5 | -0.2 | -0.2 |
| E6 | -0.6 | -0.2 |
| E7 | -1.0 | -0.2 |
| E8 | -1.4 | -0.2 |
| F1 | 1.4 | -0.6 |
| F2 | 1.0 | -0.6 |
| F3 | 0.6 | -0.6 |
| F4 | 0.2 | -0.6 |
| F5 | -0.2 | -0.6 |
| F6 | -0.6 | -0.6 |
| F7 | -1.0 | -0.6 |
| F8 | -1.4 | -0.6 |
| G1 | 1.4 | -1.0 |
| G2 | 1.0 | -1.0 |
| G3 | 0.6 | -1.0 |
| G4 | 0.2 | -1.0 |
| G5 | -0.2 | -1.0 |
| G6 | -0.6 | -1.0 |
| G7 | -1.0 | -1.0 |
| G8 | -1.4 | -1.0 |
| H1 | 1.4 | -1.4 |
| H2 | 1.0 | -1.4 |
| H3 | 0.6 | -1.4 |
| H4 | 0.2 | -1.4 |
| H5 | -0.2 | -1.4 |
| H6 | -0.6 | -1.4 |
| H7 | -1.0 | -1.4 |
| H8 | -1.4 | -1.4 |

Notes : 1. Dimension "b" is measured at the maximum ball diameter in a plane to the seating plane.

2. Applied to whole wafer.

| | TITLE | GPC | DRAWING NO. | REV. |
|---|---|---|---|---|
| Atmel<br>Package Drawing Contact:<br>packagedrawings@atmel.com | VS, 64 ball (8x8 Array), 0.4mm pitch Full Array<br>3.593 x 3.513 x 0.533mm, WLCSP | GBS | VS | A |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 59-5.** Device and Package Maximum Weight

| | |
|---|---|
| 14 | mg |

**Table 59-6.** Package Characteristics

| | |
|---|---|
| Moisture Sensitivity Level | MSL1 |

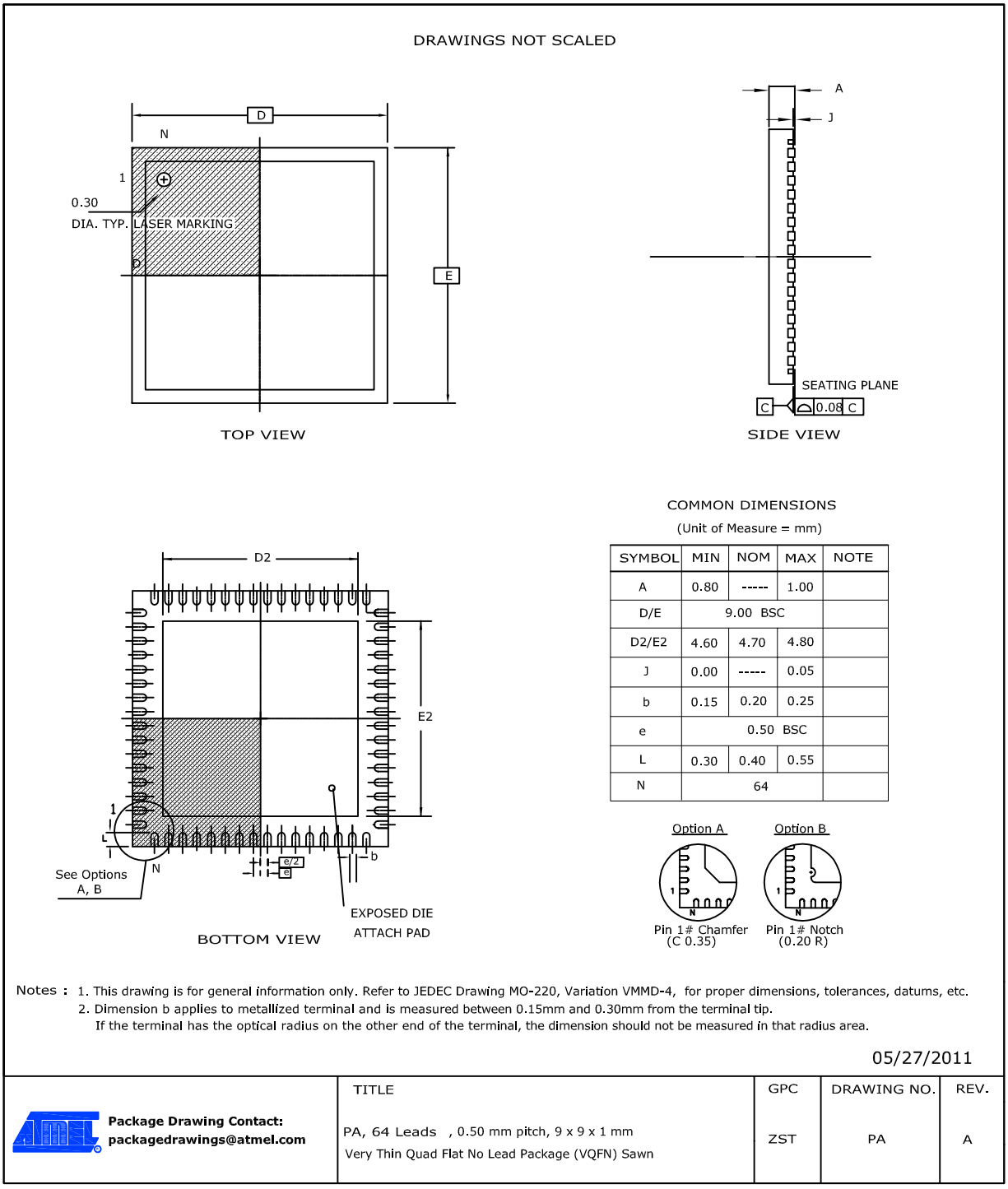
**Table 59-7.** Package Reference

| | |
|---|---|
| JEDEC Drawing Reference | N/A |
| JESD97 Classification | e1 |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 59.3.4. 64-Pin VQFN




| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | 0.80 | ----- | 1.00 | |
| D/E | | 9.00 BSC | | |
| D2/E2 | 4.60 | 4.70 | 4.80 | |
| J | 0.00 | ----- | 0.05 | |
| b | 0.15 | 0.20 | 0.25 | |
| e | | 0.50 BSC | | |
| L | 0.30 | 0.40 | 0.55 | |
| N | | 64 | | |

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MO-220, Variation VMMD-4, for proper dimensions, tolerances, datums, etc.
2. Dimension b applies to metallized terminal and is measured between 0.15mm and 0.30mm from the terminal tip.
If the terminal has the optical radius on the other end of the terminal, the dimension should not be measured in that radius area.

05/27/2011

| | TITLE | GPC | DRAWING NO. | REV. |
|---|---|---|---|---|
| Package Drawing Contact: packagedrawings@atmel.com | PA, 64 Leads , 0.50 mm pitch, 9 x 9 x 1 mm Very Thin Quad Flat No Lead Package (VQFN) Sawn | ZST | PA | A |

**Note:** The exposed die attach pad is not connected electrically inside the device. It is recommended that the exposed pad be connected to ground in the PCB.

**Table 59-8.** Device and Package Maximum Weight

| | |
|---|---|
| 200 | mg |

**Table 59-9.** Package Characteristics

| | |
|---|---|
| Moisture Sensitivity Level | MSL3 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 59-10.** Package Reference

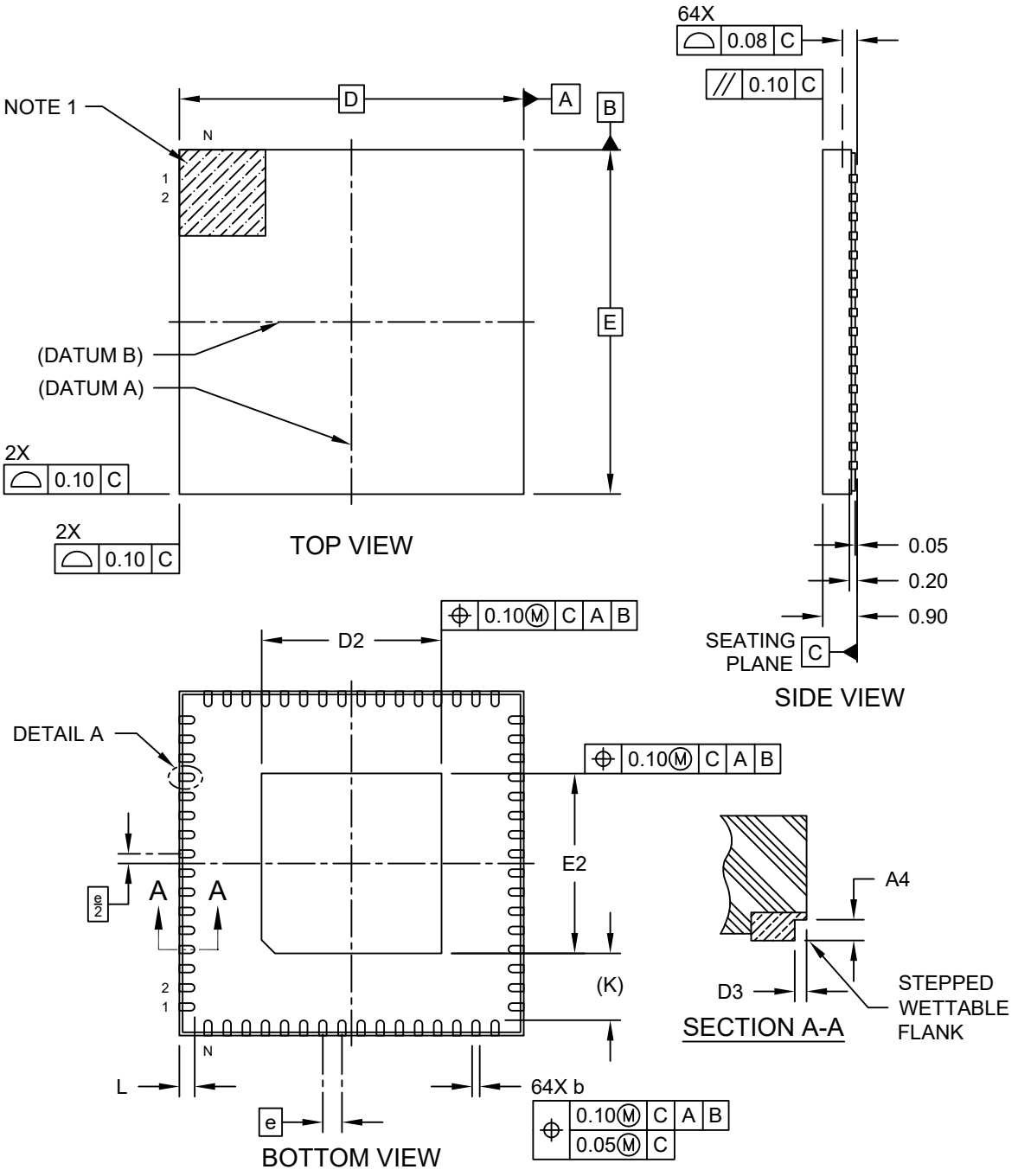
| | |
|---|---|
| JEDEC Drawing Reference | MO-220 |
| JESD97 Classification | E3 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 59.3.5. 64-Pin VQFN Wettable Flanks

**64-Lead Very Thin Plastic Quad Flat, No Lead Package (U6B) - 9x9 mm Body [VQFN] With 4.7 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZRB**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|

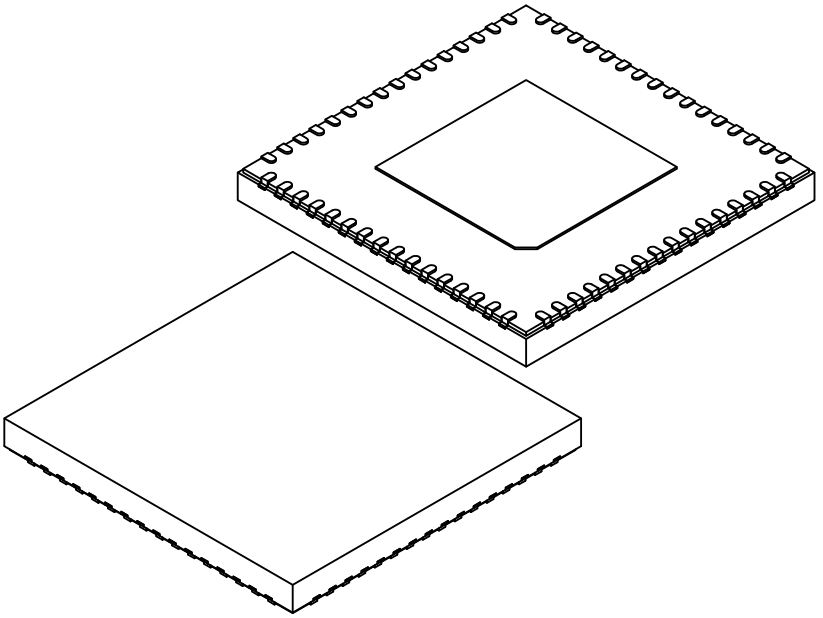



Microchip Technology Drawing C04-21497 Rev A Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Very Thin Plastic Quad Flat, No Lead Package (U6B) - 9x9 mm Body [VQFN] With 4.7 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZRB

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



DETAIL 1
ALTERNATE TERMINAL CONFIGURATIONS

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Terminals | N | | 64 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | 0.80 | 0.85 | 0.90 |
| Standoff | A1 | 0.00 | 0.035 | 0.05 |
| Terminal Thickness | A3 | | 0.203 REF | |
| Overall Length | D | | 9.00 BSC | |
| Exposed Pad Length | D2 | 4.60 | 4.70 | 4.80 |
| Overall Width | E | | 9.00 BSC | |
| Exposed Pad Width | E2 | 4.60 | 4.70 | 4.80 |
| Terminal Width | b | 0.15 | 0.20 | 0.25 |
| Terminal Length | L | 0.35 | 0.40 | 0.45 |
| Terminal-to-Exposed-Pad | K | | 1.75 REF | |
| Wettable Flank Step Length | D3 | - | - | 0.085 |
| Wettable Flank Step Height | A4 | 0.10 | - | 0.19 |

Notes:

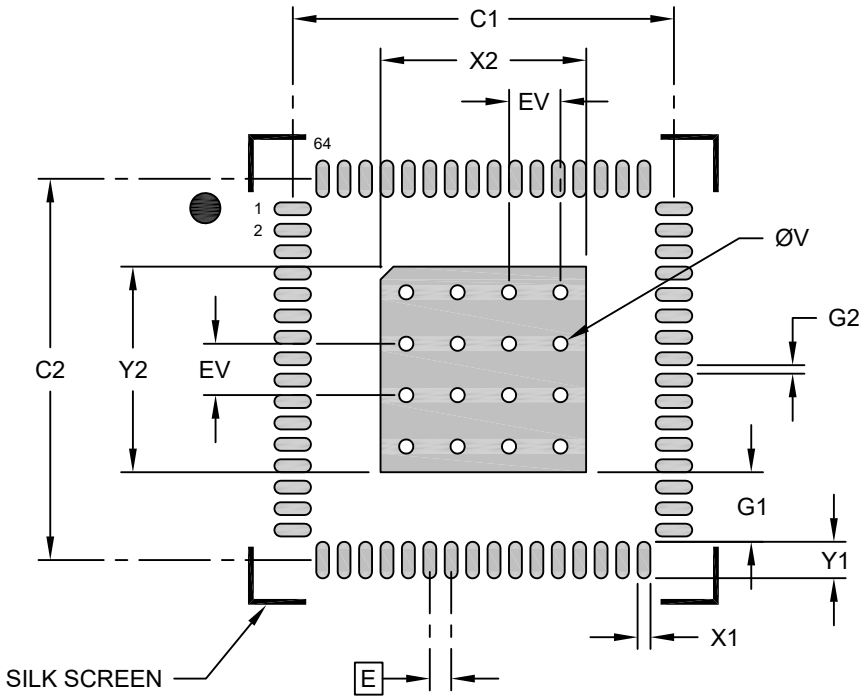
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21497 Rev A Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Very Thin Plastic Quad Flat, No Lead Package (U6B) - 9x9 mm Body [VQFN] With 4.7 mm Exposed Pad and Stepped Wettable Flanks; Atmel Legacy ZRB

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Optional Center Pad Width | X2 | | | 4.80 |
| Optional Center Pad Length | Y2 | | | 4.80 |
| Contact Pad Spacing | C1 | | 8.90 | |
| Contact Pad Spacing | C2 | | 8.90 | |
| Contact Pad Width (X64) | X1 | | | 0.30 |
| Contact Pad Length (X64) | Y1 | | | 0.85 |
| Contact Pad to Center Pad (X64) | G1 | 1.63 | | |
| Contact Pad to Contact Pad (X60) | G2 | 0.20 | | |
| Thermal Via Diameter | V | | 0.33 | |
| Thermal Via Pitch | EV | | 1.20 | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

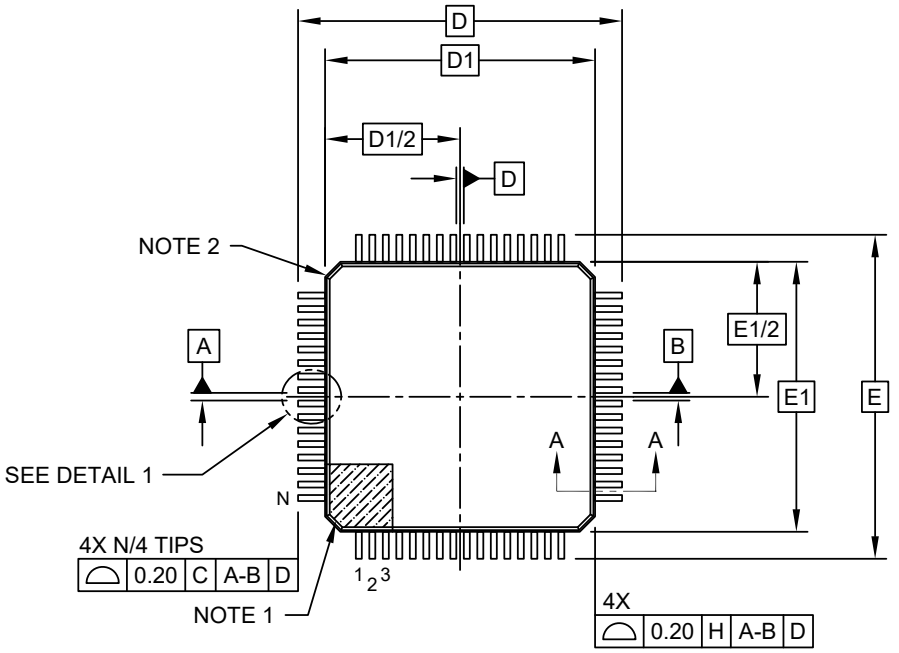
Microchip Technology Drawing C04-23497 Rev A


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

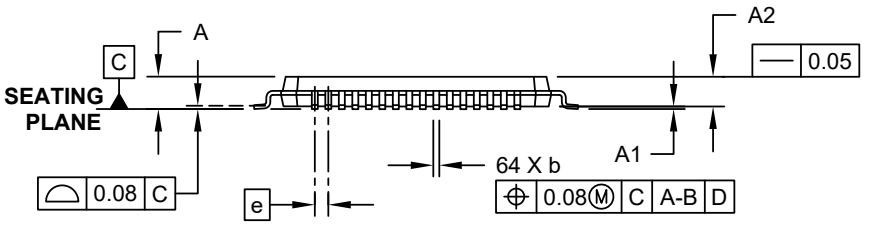
### 59.3.6. 64-pin TQFP

**64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|




TOP VIEW



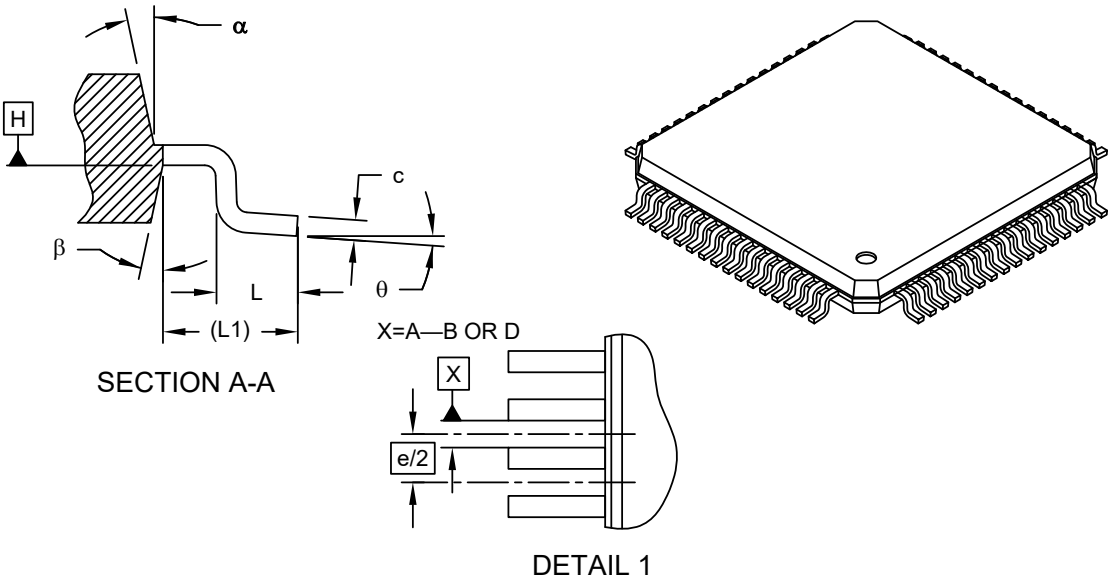

SIDE VIEW

Microchip Technology Drawing C04-085C Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Leads | N | | 64 | |
| Lead Pitch | e | | 0.50 BSC | |
| Overall Height | A | - | - | 1.20 |
| Molded Package Thickness | A2 | 0.95 | 1.00 | 1.05 |
| Standoff | A1 | 0.05 | - | 0.15 |
| Foot Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | L1 | | 1.00 REF | |
| Foot Angle | ϕ | 0° | 3.5° | 7° |
| Overall Width | E | | 12.00 BSC | |
| Overall Length | D | | 12.00 BSC | |
| Molded Package Width | E1 | | 10.00 BSC | |
| Molded Package Length | D1 | | 10.00 BSC | |
| Lead Thickness | c | 0.09 | - | 0.20 |
| Lead Width | b | 0.17 | 0.22 | 0.27 |
| Mold Draft Angle Top | α | 11° | 12° | 13° |
| Mold Draft Angle Bottom | β | 11° | 12° | 13° |

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

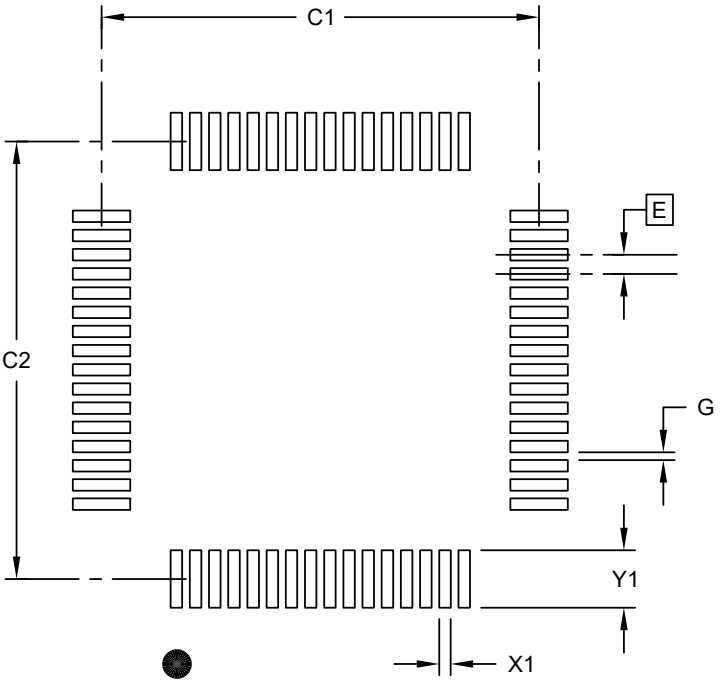
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085C Sheet 2 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Plastic Thin Quad Flatpack (PT)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Contact Pad Spacing | C1 | | 11.40 | |
| Contact Pad Spacing | C2 | | 11.40 | |
| Contact Pad Width (X28) | X1 | | | 0.30 |
| Contact Pad Length (X28) | Y1 | | | 1.50 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

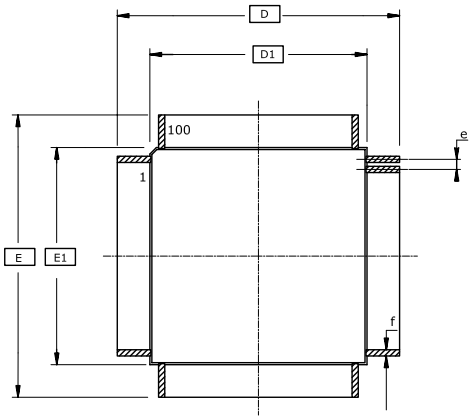
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-2085B Sheet 1 of 1

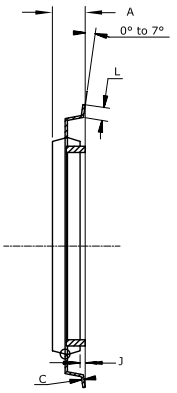

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 59.3.7. 100 pin TQFP

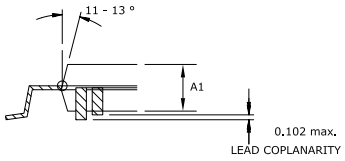
DRAWINGS NOT SCALED




TOP VIEW




SIDE VIEW




DETAIL VIEW

COMMON DIMENSIONS

(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | ------ | ------ | 1.20 | |
| A1 | 0.95 | ----- | 1.05 | |
| C | 0.09 | ----- | 0.20 | |
| D/E | | 16.00 | | |
| D1/E1 | | 14.00 | | 2 |
| J | 0.05 | ----- | 0.15 | |
| L | 0.45 | ----- | 0.75 | |
| e | | 0.50 BSC | | |
| f | 0.17 | ----- | 0.27 | |
| n | | 100 | | |

Notes : 1. This drawing is for general information only. Refer to JEDEC Drawing MS-026, Variation AED.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

**Table 59-11.** Device and Package Maximum Weight

| | |
|---|---|
| 520 | mg |

**Table 59-12.** Package Characteristics

| | |
|---|---|
| Moisture Sensitivity Level | MSL3 |

**Table 59-13.** Package Reference

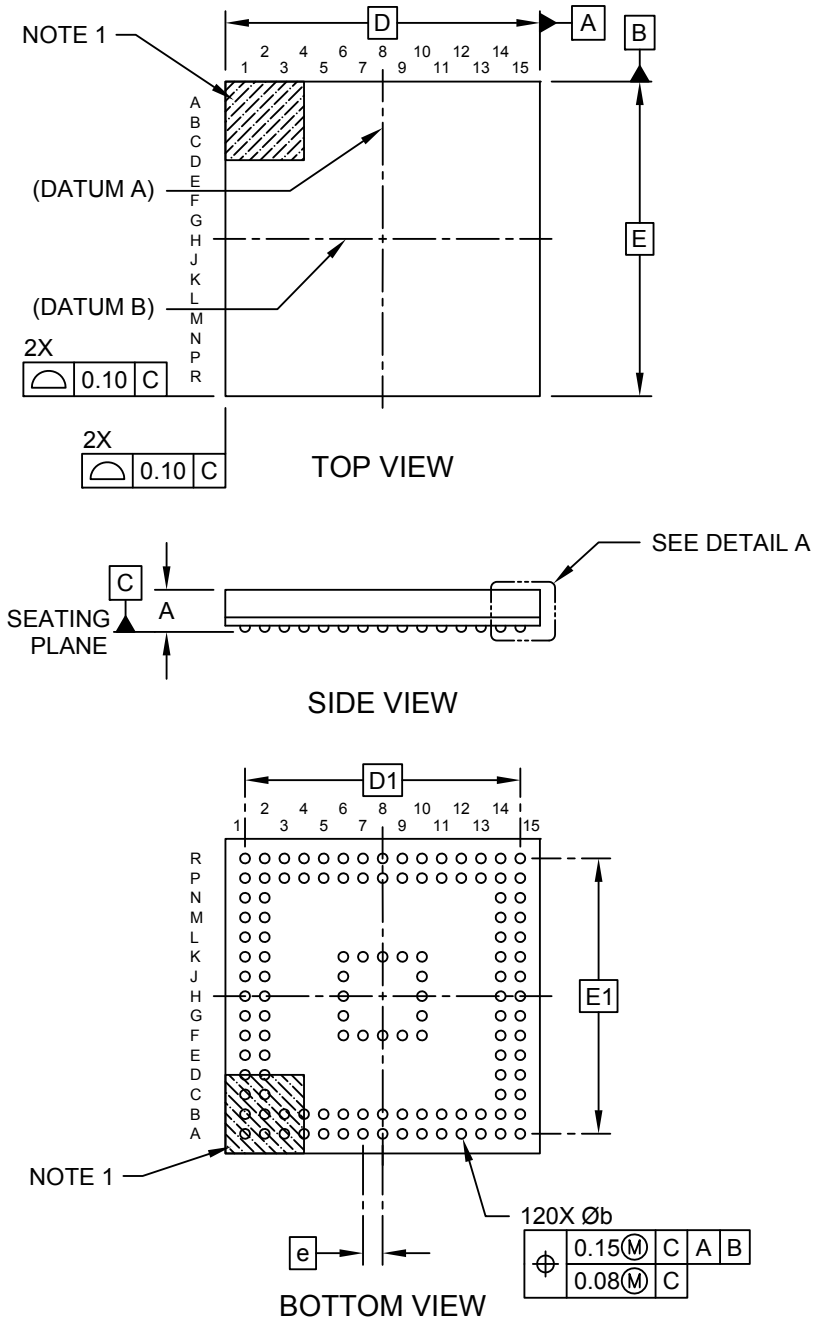
| | |
|---|---|
| JEDEC Drawing Reference | MS-026, variant AED |
| JESD97 Classification | e3 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 59.3.8. 120-ball TFBGA

**120-Ball Thin Fine Pitch Ball Grid Array Package (DGB) - 8x8 mm Body [TFBGA]**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|

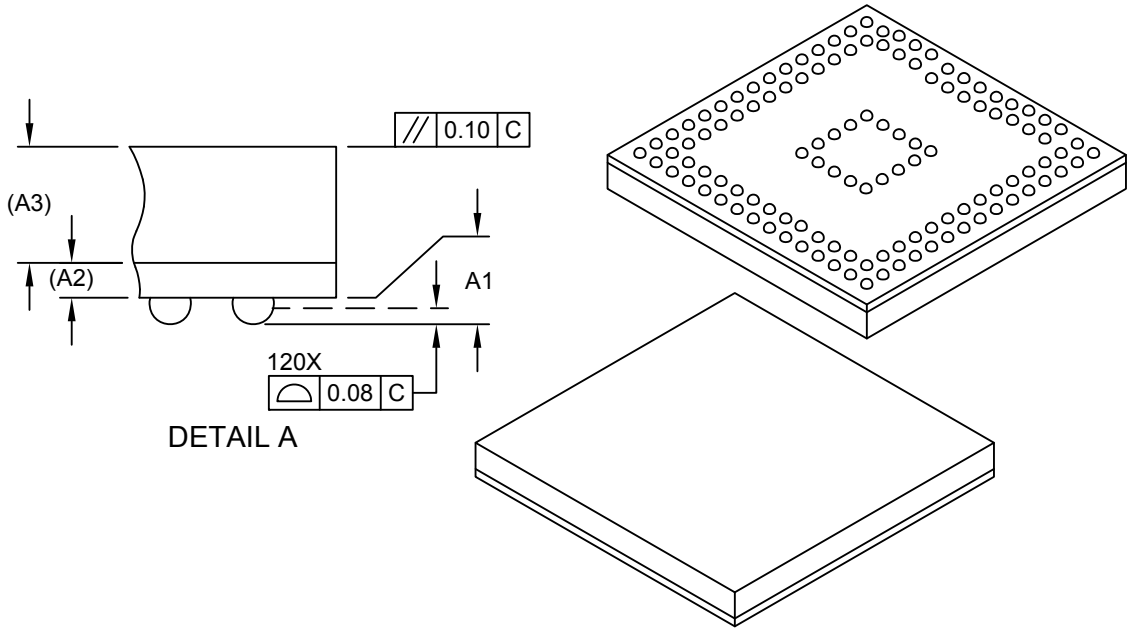



Microchip Technology Drawing C04-21465-DGB Rev B Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 120-Ball Thin Fine Pitch Ball Grid Array Package (DGB) - 8x8 mm Body [TFBGA]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Terminals | N | | 120 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | - | - | 1.20 |
| Standoff | A1 | 0.11 | - | 0.21 |
| Substrate Thickness | A2 | | 2.10 REF | |
| Mold Cap Thickness | A3 | | 0.70 REF | |
| Overall Length | D | | 8.00 BSC | |
| Overall Ball Pitch | D1 | | 7.00 BSC | |
| Overall Width | E | | 8.00 BSC | |
| Exposed Pad Width | E1 | | 7.00 BSC | |
| Terminal Width | b | 0.20 | - | 0.30 |

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

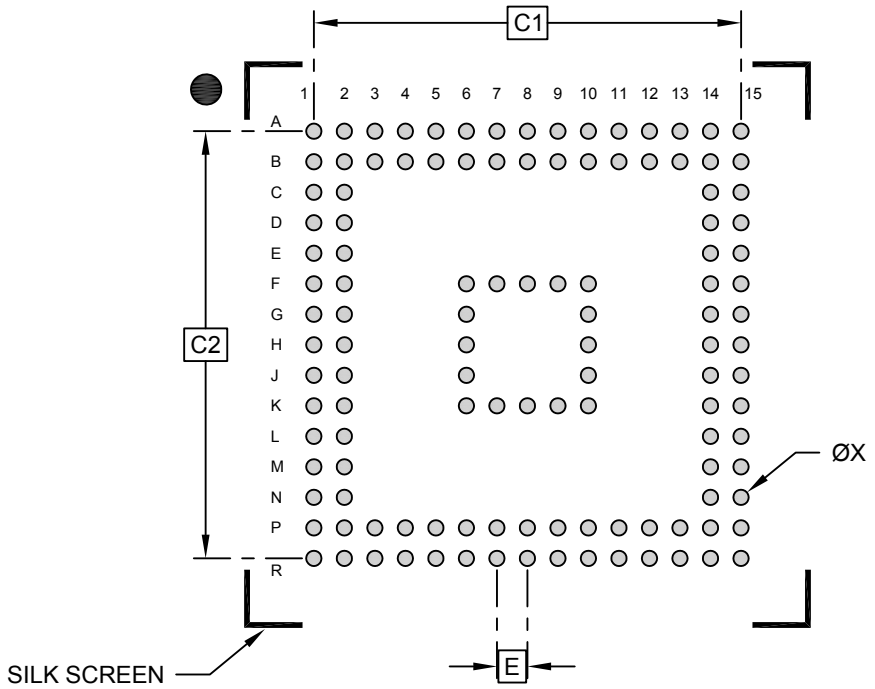
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21465-DGB Rev B Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 120-Ball Thin Fine Pitch Ball Grid Array Package (DGB) - 8x8 mm Body [TFBGA]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Contact Pad Spacing | C1 | | 7.00 BSC | |
| Contact Pad Spacing | C2 | | 7.00 BSC | |
| Contact Pad Width (X20) | X | | 0.25 | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

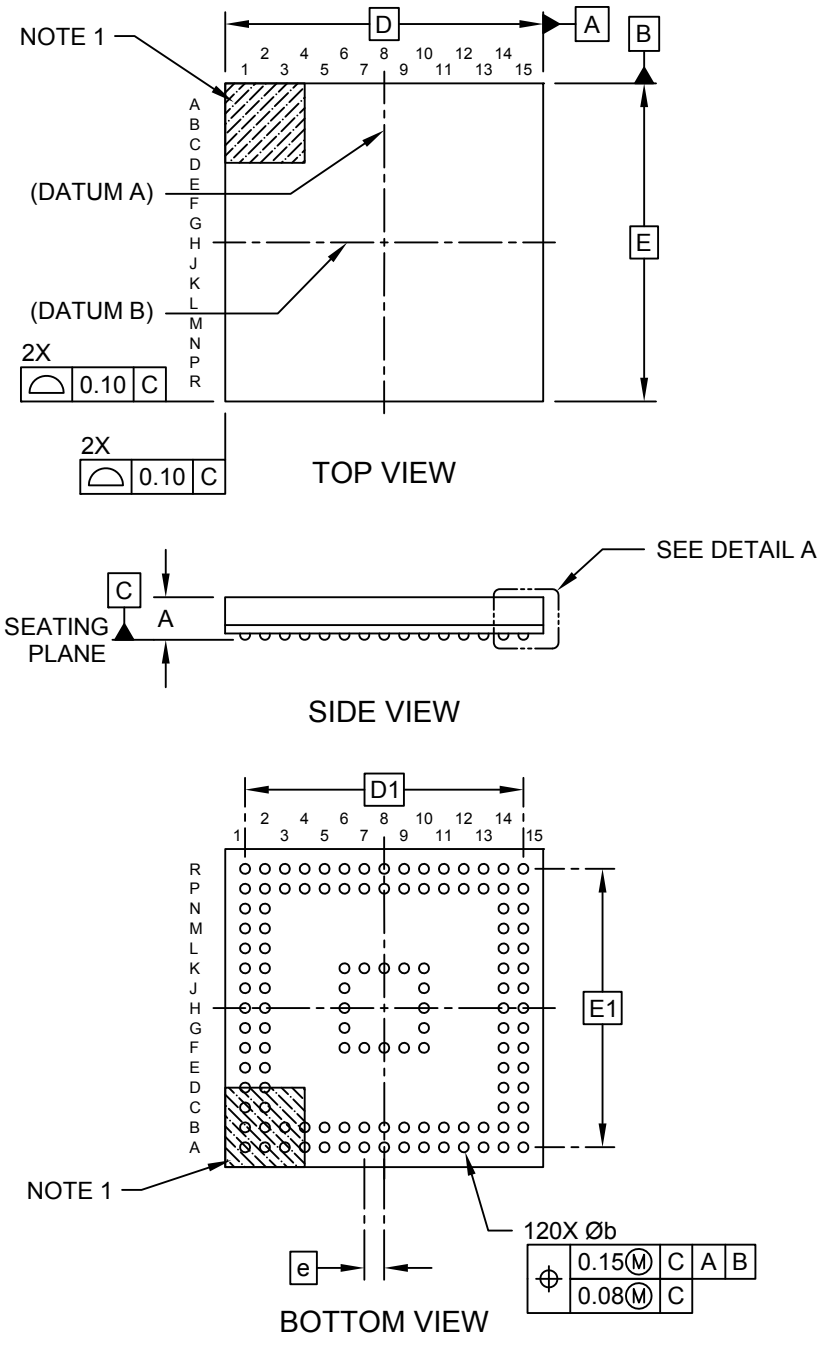
Microchip Technology Drawing C04-23465-DGB Rev B


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 59.3.9. 120-ball TFBGA (AEC-Q100 Grade 1)

**120-Ball Thin Fine Pitch Ball Grid Array Package (HEB) - 8x8 mm Body [TFBGA]**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|

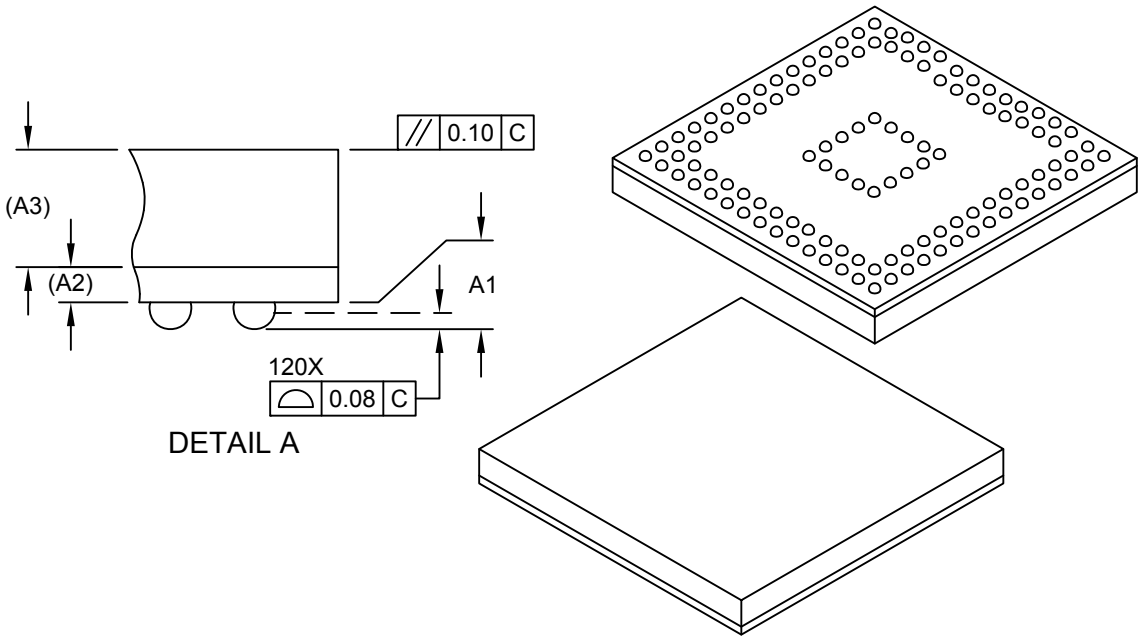



Microchip Technology Drawing C04-21465-HEB Rev B Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 120-Ball Thin Fine Pitch Ball Grid Array Package (HEB) - 8x8 mm Body [TFBGA]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Terminals | N | | 120 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | - | - | 1.20 |
| Standoff | A1 | 0.11 | - | 0.21 |
| Substrate Thickness | A2 | | 2.10 REF | |
| Mold Cap Thickness | A3 | | 0.70 REF | |
| Overall Length | D | | 8.00 BSC | |
| Overall Ball Pitch | D1 | | 7.00 BSC | |
| Overall Width | E | | 8.00 BSC | |
| Exposed Pad Width | E1 | | 7.00 BSC | |
| Terminal Width | b | 0.20 | - | 0.30 |

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

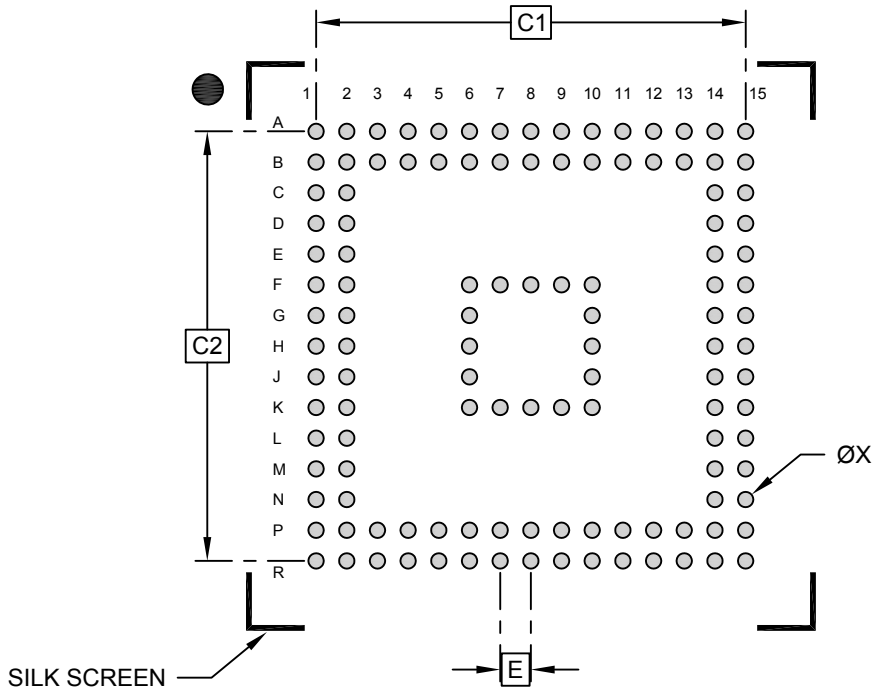
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-21465-HEB Rev B Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 120-Ball Thin Fine Pitch Ball Grid Array Package (HEB) - 8x8 mm Body [TFBGA]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




RECOMMENDED LAND PATTERN

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Contact Pitch | E | | 0.50 BSC | |
| Contact Pad Spacing | C1 | | 7.00 BSC | |
| Contact Pad Spacing | C2 | | 7.00 BSC | |
| Contact Pad Width (X20) | X | | 0.25 | |

Notes:

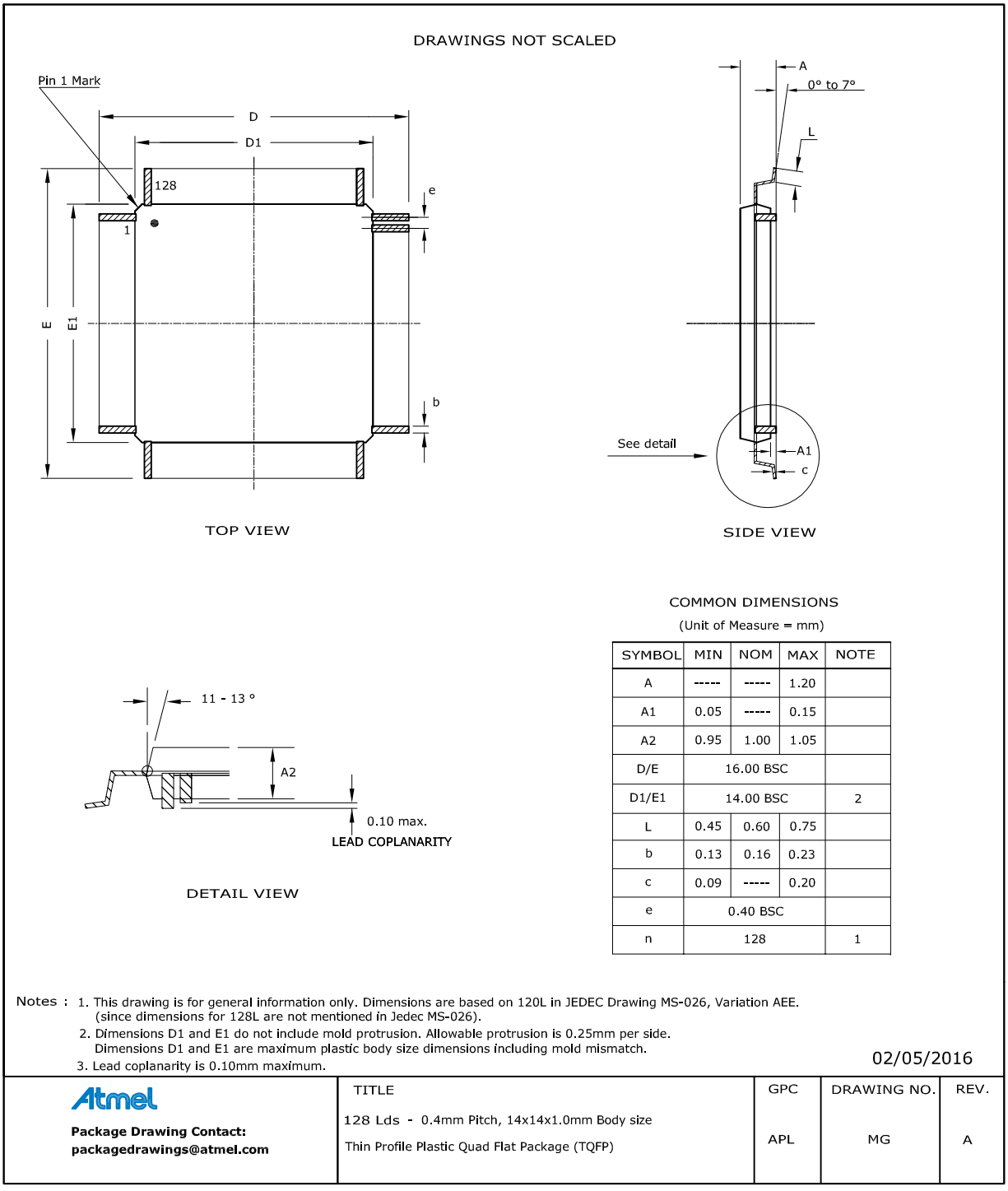
1. Dimensioning and tolerancing per ASME Y14.5M

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-23465-HEB Rev B


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 59.3.10. 128 pin TQFP




COMMON DIMENSIONS

(Unit of Measure = mm)

| SYMBOL | MIN | NOM | MAX | NOTE |
|---|---|---|---|---|
| A | ----- | ----- | 1.20 | |
| A1 | 0.05 | ----- | 0.15 | |
| A2 | 0.95 | 1.00 | 1.05 | |
| D/E | | 16.00 BSC | | |
| D1/E1 | | 14.00 BSC | | 2 |
| L | 0.45 | 0.60 | 0.75 | |
| b | 0.13 | 0.16 | 0.23 | |
| c | 0.09 | ----- | 0.20 | |
| e | | 0.40 BSC | | |
| n | | 128 | | 1 |

Notes :
1. This drawing is for general information only. Dimensions are based on 120L in JEDEC Drawing MS-026, Variation AEE. (since dimensions for 128L are not mentioned in Jedec MS-026).
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.10mm maximum.

02/05/2016

| Atmel | TITLE | GPC | DRAWING NO. | REV. |
|---|---|---|---|---|
| **Package Drawing Contact:** **packagedrawings@atmel.com** | 128 Lds - 0.4mm Pitch, 14x14x1.0mm Body size Thin Profile Plastic Quad Flat Package (TQFP) | APL | MG | A |

**Table 59-14.** Device and Package Maximum Weight

| | |
|---|---|
| 520 | mg |

**Table 59-15.** Package Characteristics

| | |
|---|---|
| Moisture Sensitivity Level | MSL3 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 59-16.** Package Reference

| | |
|---|---|
| JEDEC Drawing Reference | MS-026 |
| JESD97 Classification | E3 |

## 59.4. Soldering Profile

The following table gives the recommended soldering profile from J-STD-20.

**Table 59-17.** Recommended Soldering Profile

| Profile Feature | Green Package |
|---|---|
| Average Ramp-up Rate (217°C to peak) | 3°C/s max. |
| Preheat Temperature 175°C ±25°C | 150-200°C |
| Time Maintained Above 217°C | 60-150s |
| Time within 5°C of Actual Peak Temperature | 30s |
| Peak Temperature Range | 260°C |
| Ramp-down Rate | 6°C/s max. |
| Time 25°C to Peak Temperature | 8 minutes max. |

A maximum of three reflow passes is allowed per component.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 60. Schematic Checklist

## 60.1. Introduction

This chapter describes a common checklist which must be used when starting and reviewing the schematics for a SAM D5x/E5x design. It also illustrates recommended power supply connections, how to connect external analog references, programmer, debugger, oscillator, and crystal.

The chapter provides recommendations regarding low-power design constraints to minimize power consumption.

This information is not intended to be exhaustive. Its objective is to cover as many configurations of use as possible. Refer to the *"Basic 32-Bit MCU Design and Troubleshooting Checklist"* guide on the Microchip web site, www.microchip.com/.

### 60.1.1. Operation in Noisy Environment

If the device is operating in an environment with much electromagnetic noise, it must be protected from this noise to ensure reliable operation. In addition to following best practice EMC design guidelines, the recommendations listed in the schematic checklist sections must be followed. In particular, placing decoupling capacitors very close to the power pins, an RC-filter on the $\overline{\text{RESET}}$ pin, and a pull-up resistor on the SWCLK pin is critical for reliable operations. It is also relevant to eliminate or attenuate noise in order to avoid that it reaches supply pins, I/O pins and crystals.

**Important:** Refer to the *EMI, EMC, EFT, and ESD Circuit Design Consideration for 32-bit Microcontrollers* guide on the Microchip web site, www.microchip.com/.

## 60.2. Power Supply

The SAM D5x/E5x supports a single or dual power supply from 1.71V to 3.63V. The same voltage must be applied to both VDDIO and VDDANA. VDDIOB level must be lower or equal to VDDIO / VDDANA.

When I/O pads in the VDDIOB cluster are multiplexed as analog pads, VDDANA is used to power the I/O. Using this configuration may result in an electrical conflict if the VDDIOB voltage is different from that of VDDIO / VDDANA. If the application has such requirements, it is required to power VDDIOB, VDDIO, and VDDANA from the same supply source to ensure that they are always at the same voltage.

The internal voltage regulator has four different modes:

- Linear mode: This mode does not require any external inductor. This is the default mode when CPU and peripherals are running
- Switching mode (Buck): The most efficient mode when the CPU and peripherals are running.
- Low Power (LP) mode: This is the default mode used when the device is in Standby mode
- Shutdown mode: When the device is in Backup mode, the internal regulator is turned off

Selecting between switching mode and linear mode can be done by software on the fly, but the power supply must be designed according to which mode is to be used.

### 60.2.1. Power Supply Connections

The following figures shows the recommended power supply connections for switched/linear mode, linear mode only and with battery backup.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 60-1.** Power Supply Connection for Switching/Linear Mode

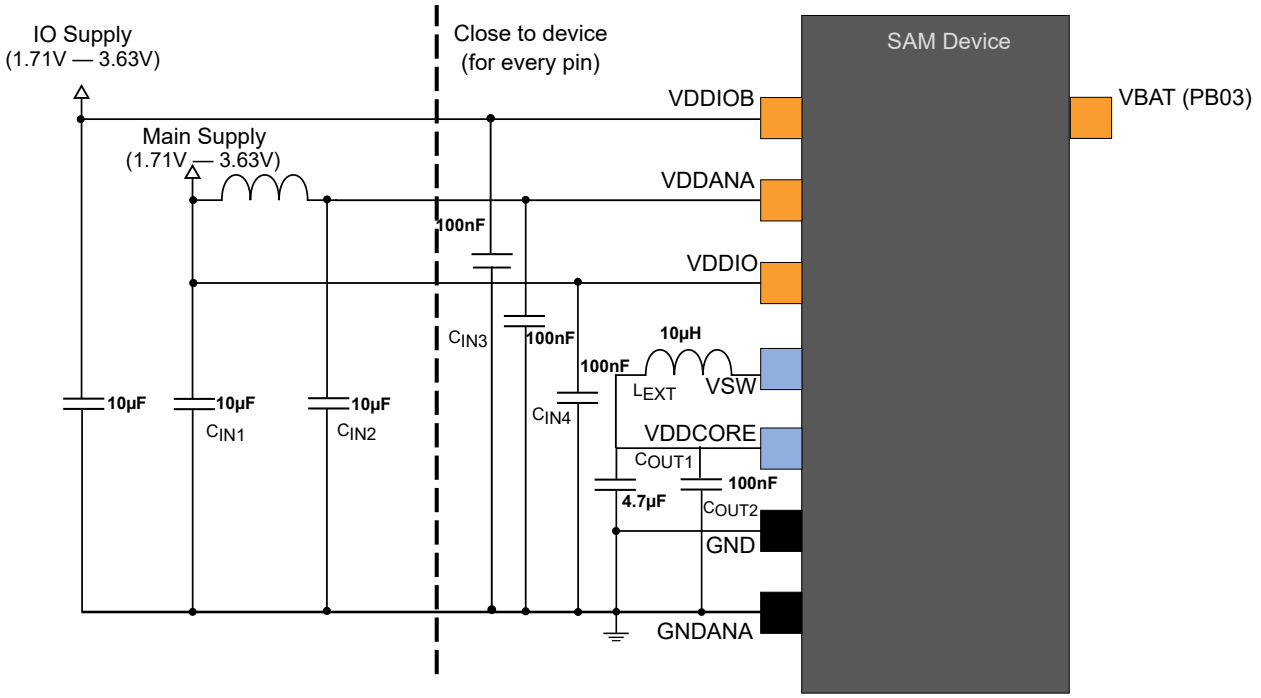
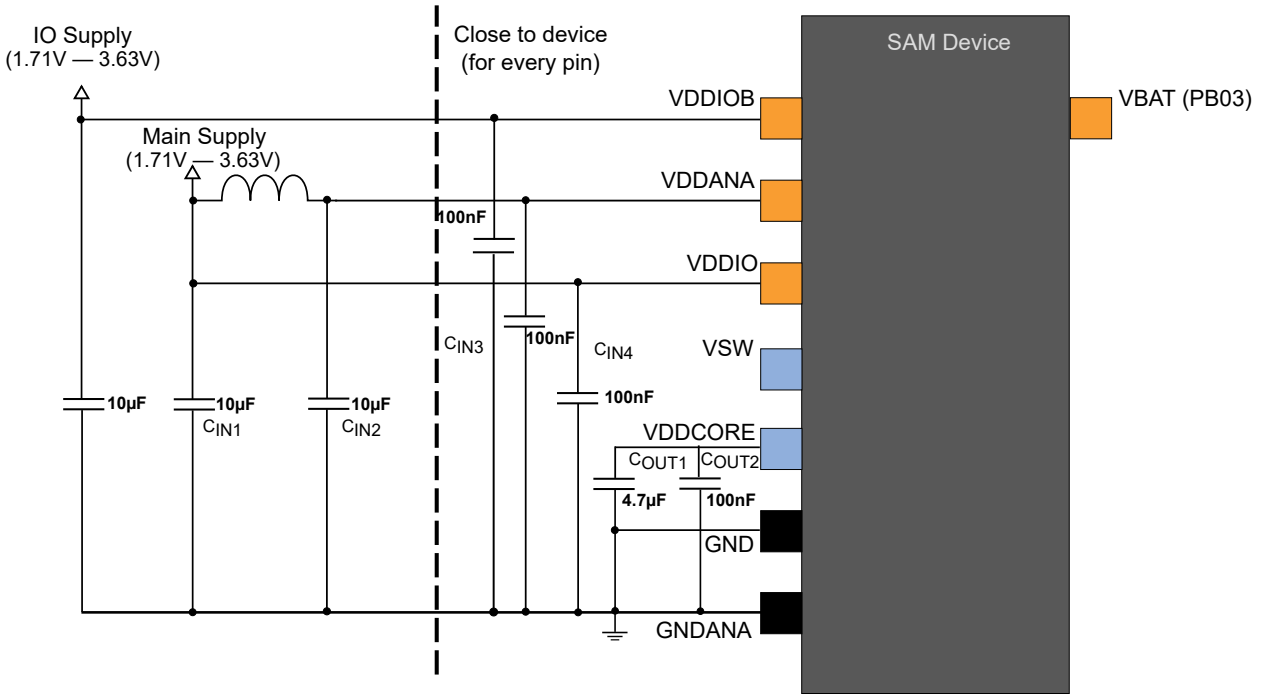



**Figure 60-2.** Power Supply Connection for Linear Mode Only

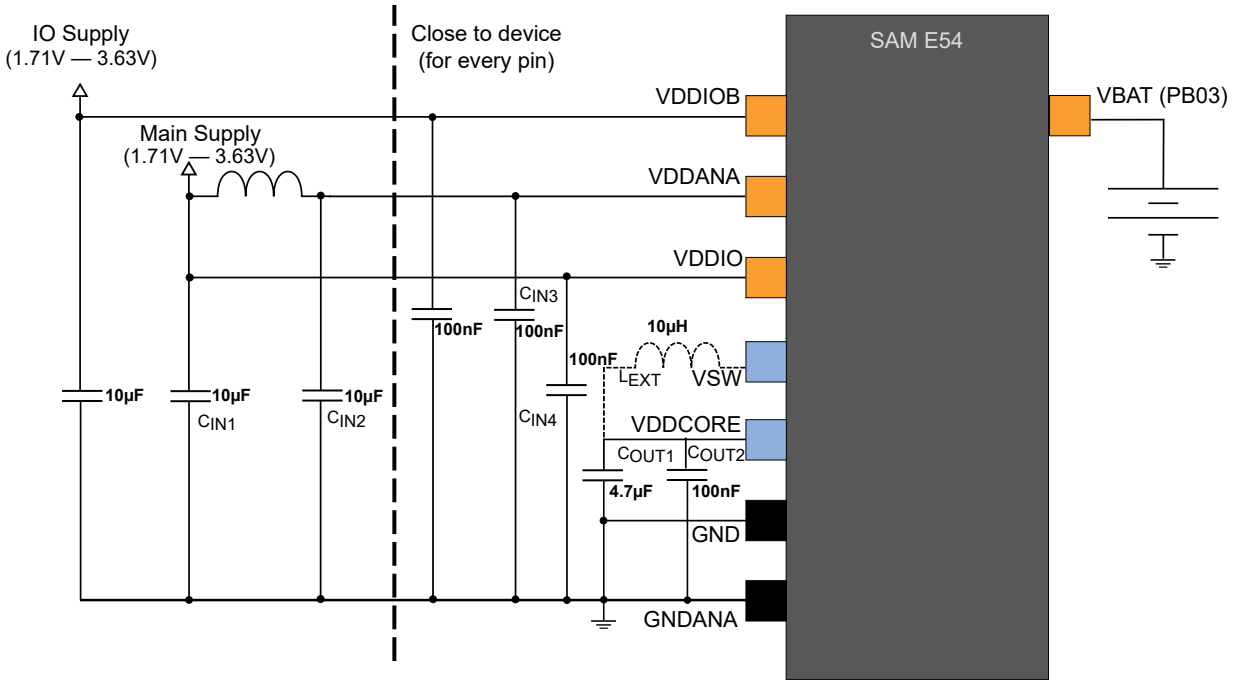




© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

**Figure 60-3.** Power Supply Connection for Battery Backup




**Notes:**

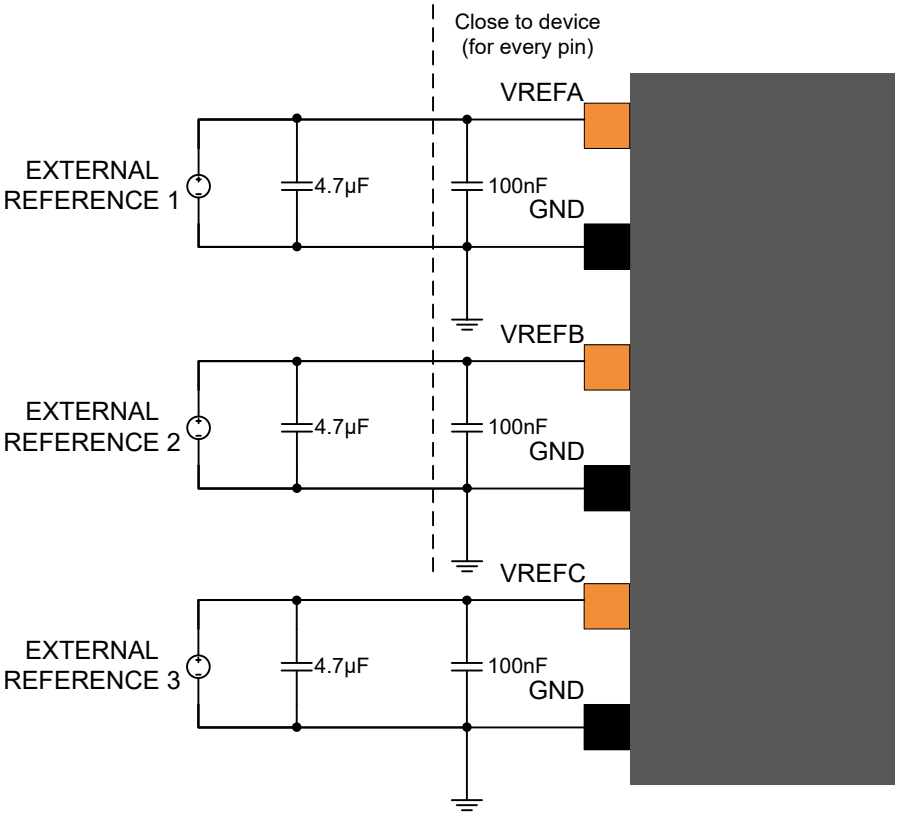
1. The passive component value shown in figures 58-1, 58-2 & 58-3 is a typical example. Refer to 54.10.1 Voltage Regulator Characteristics for details on specification.
2. Decoupling capacitors should be placed close to the device for each supply pin pair in the signal group, low ESR capacitors should be used for better decoupling.
3. An inductor should be added between the external power and the $V_{DD}$ for power filtering.
4. A ferrite bead has better filtering performance compared to standard inductor at high frequencies. A ferrite bead can be added between the main power supply ($V_{DD}$) and $V_{DDANA}$ to prevent digital noise from entering the analog power domain. The bead should provide enough impedance (i.e., 50Ω at 20 MHz and 220Ω at 100 MHz) to separate the digital and analog power domains. Make sure to select a ferrite bead designed for filtering applications with a low DC resistance to avoid a large voltage drop across the ferrite bead.

## 60.3. External Analog Reference Connections

The following schematic checklist is only necessary if the application is using one or more of the external analog references. If the internal references are used instead, the following circuits are not necessary.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

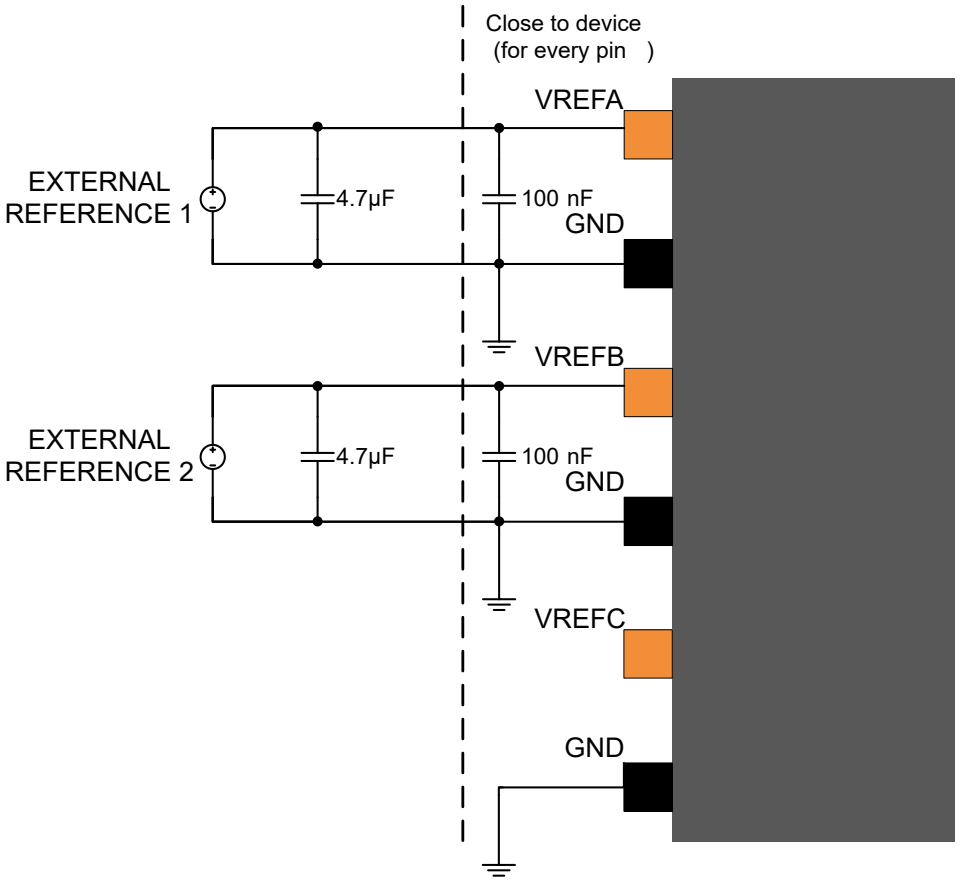
**Figure 60-4.** External Analog Reference Schematic With Three References





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 60-5.** External Analog Reference Schematic With Two References





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 60-6.** External Analog Reference Schematic With One Reference

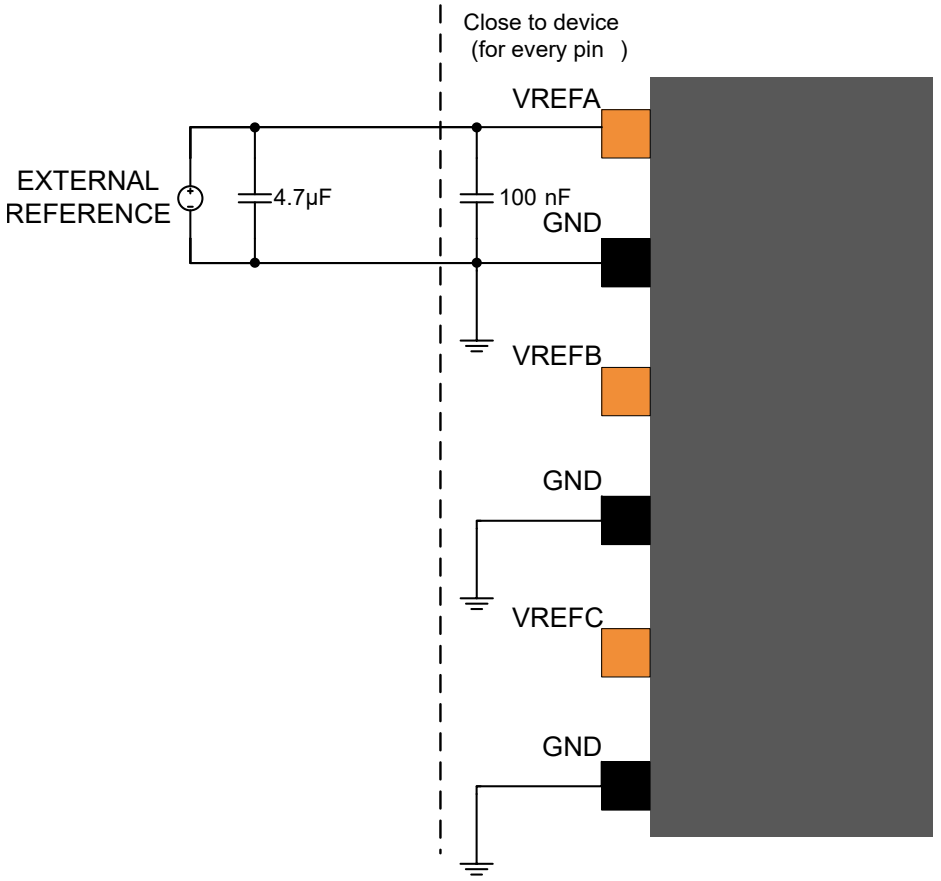



**Table 60-1.** External Analog Reference Connections

| Signal Name | Recommended Pin Connection | Description |
|---|---|---|
| VREFx | 1.0V to ($V_{DDANA}$ - 0.4V) for ADC<br>1.0V to ($V_{DDANA}$ - 0.4V) for DAC<br>Decoupling/filtering capacitors<br>100nF(1)(2) and 4.7µF(1) | External reference VREFx for the analog port |
| GND | | Ground |

**Notes:**

1. These values are only given as a typical example.
2. Decoupling capacitor should be placed close to the device for each supply pin pair in the signal group.

## 60.4. External Reset Circuit

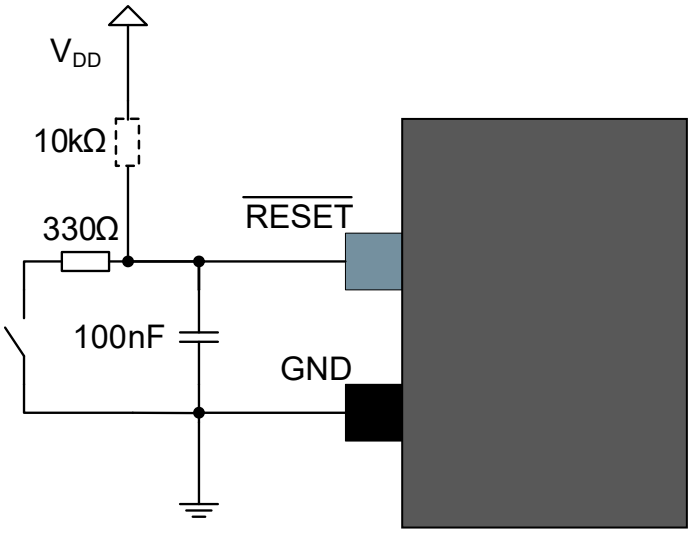
When the external Reset function is used, connect the external Reset circuit to the $\overline{\text{RESET}}$ pin as shown below. If the external Reset function is not required, the circuit is not necessary. The $\overline{\text{RESET}}$ pin can either remain unconnected, or be driven LOW externally by the application circuitry.

The Reset switch can also be removed if a manual Reset is not necessary. The $\overline{\text{RESET}}$ pin itself has an internal pull-up resistor, therefore it is optional to add any external pull-up resistor.

A pull-up resistor makes sure that the Reset does not go low and unintentionally causing a device Reset. An additional resistor has been added in series with the switch to safely discharge the filtering capacitor, that is, preventing a current surge when shorting the filtering capacitor which again can cause a noise spike that can have a negative effect on the system.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 60-7.** External Reset Circuit Schematic




**Note:** Reset circuit recommendations to improve EFT/ESD/EMI immunity can be found in the *"Basic 32-Bit MCU Design and Troubleshooting Checklist"* guide avaialbe on the Microchip web site www.microchip.com/.

**Table 60-2.** Reset Circuit Connections

| Signal Name | Recommended Pin Connection | Description |
|---|---|---|
| $\overline{\text{RESET}}$ | Reset low level threshold voltage<br>$V_{DDIO}$ = 1.71V - 2.0V: Below 0.33 * $V_{DDIO}$<br>$V_{DDIO}$ = 2.7V - 3.6V: Below 0.36 * $V_{DDIO}$<br>Decoupling/filter capacitor 100nF(1)<br>Pull-up resistor 10kΩ(1,2)<br>Resistor in series with the switch 330Ω(1) | Reset pin |

1. These values are only given as a typical example.

2. The SAM D5x/E5x features an internal pull-up resistor on the $\overline{\text{RESET}}$ pin, hence an external pull-up is optional.

## 60.5. Unused or Unconnected Pins

For unused pins the default state of the pins will give the lowest current leakage. Thus there is no need to do any configuration of the unused pins in order to lower the power consumption.

## 60.6. Clocks and Crystal Oscillators

The SAM D5x/E5x can be run from internal or external clock sources, or a mix of internal and external sources. An example of usage can be to use the internal 48MHz DFLL as source for the system clock and an external 32.768kHz watch crystal as clock source for the Real-Time counter (RTC).


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

### 60.6.1. External Clock Source

**Figure 60-8.** External Clock Source Schematic

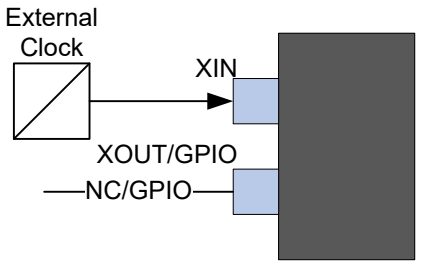
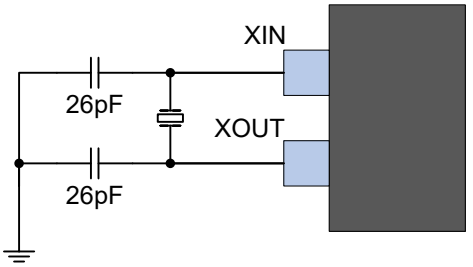



**Table 60-3.** External Clock Source Connections

| Signal Name | Recommended Pin Connection | Description |
|---|---|---|
| XIN | XIN is used as input for an external clock signal | Input for inverting oscillator pin |
| XOUT/GPIO | Can be left unconnected or used as normal GPIO | NC/GPIO |

### 60.6.2. Crystal Oscillator

**Figure 60-9.** Crystal Oscillator Schematic




The crystal should be located as close to the device as possible. Long signal lines may cause too high load to operate the crystal, and cause crosstalk to other parts of the system.

**Table 60-4.** Crystal Oscillator Checklist

| Signal Name | Recommended Pin Connection | Description |
|---|---|---|
| XIN | Load capacitor 26pF(1)(2) | External crystal between 8 to 48MHz |
| XOUT | Load capacitor 26pF(1)(2) | |

1. These values are only given as a typical example.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

### 60.6.3. External Real Time Oscillator

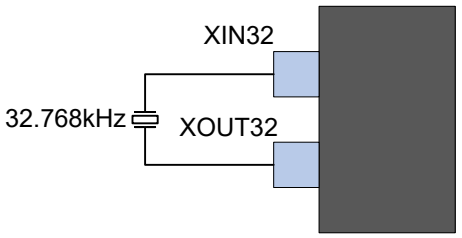
The low frequency crystal oscillator is optimized for use with a 32.768kHz watch crystal. When selecting crystals, load capacitance and the crystal's Equivalent Series Resistance (ESR) must be taken into consideration. Both values are specified by the crystal vendor.

SAM D5x/E5x oscillator is optimized for very low power consumption, hence close attention should be made when selecting crystals.

The typical parasitic load capacitance values are available in the Electrical Characteristics section. This capacitance and PCB capacitance can allow using a crystal inferior to 12.5pF load capacitance without external capacitors as shown in the next figure.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 60-10.** External Real Time Oscillator without Load Capacitor




To improve accuracy and Safety Factor, the crystal datasheet can recommend adding external capacitors as shown the figure below.

To find suitable load capacitance for a 32.768kHz crystal, consult the crystal datasheet.

**Figure 60-11.** External Real Time Oscillator with Load Capacitor

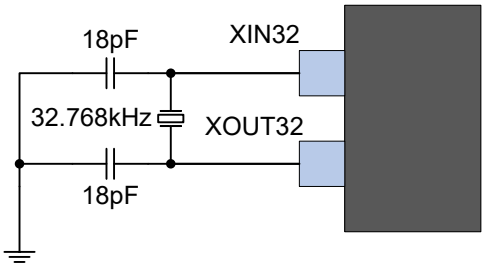



**Table 60-5.** External Real Time Oscillator Checklist

| Signal Name | Recommended Pin Connection | Description |
|---|---|---|
| XIN32 | Load capacitor 18pF(1)(2) | Timer oscillator input |
| XOUT32 | Load capacitor 18pF(1)(2) | Timer oscillator output |

1. These values are only given as typical examples.

2. The capacitors should be placed close to the device for each supply pin pair in the signal group.

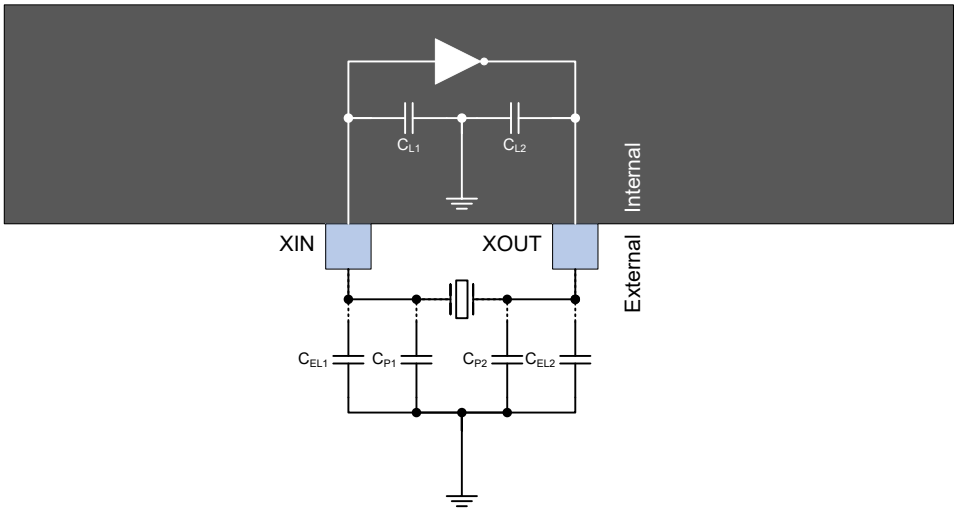
**Note:** In order to minimize the cycle-to-cycle jitter of the external oscillator, keep the neighboring pins as steady as possible. For neighboring pin details, refer to the Oscillator Pinout section.

### 60.6.4. Calculating the Correct Crystal Decoupling Capacitor

The model shown in Figure 60-12 can be used to calculate correct load capacitor for a given crystal. This model includes internal capacitors $C_{Ln}$, external parasitic capacitance $C_{ELn}$ and external load capacitance $C_{Pn}$.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 60-12.** Crystal Circuit With Internal, External and Parasitic Capacitance




Using this model the total capacitive load for the crystal can be calculated as shown in the equation below:

$$\sum C_{\text{tot}} = \frac{(C_{L1} + C_{P1} + C_{\text{EL1}})(C_{L2} + C_{P2} + C_{\text{EL2}})}{C_{L1} + C_{P1} + C_{\text{EL1}} + C_{L2} + C_{P2} + C_{\text{EL2}}}$$

where $C_{tot}$ is the total load capacitance seen by the crystal. This value should be equal to the load capacitance value found in the crystal manufacturer datasheet.

The parasitic capacitance $C_{ELn}$ can in most applications be disregarded as these are usually very small. If accounted for, these values are dependent on the PCB material and PCB layout.

For some crystal the internal capacitive load provided by the device itself can be enough. To calculate the total load capacitance in this case. $C_{ELn}$ and $C_{Pn}$ are both zero, $C_{L1} = C_{L2} = C_L$, and the equation reduces to the following:

$$\sum C_{\text{tot}} = \frac{C_L}{2}$$

See the related links for equivalent internal pin capacitance values.

## 60.7. Programming and Debug Ports

For programming and debugging the SAM D5x/E5x, the device must be connected using the Serial Wire Debug (SWD) and interface. Currently the SWD interface is supported by several Microchip and third party programmers and debuggers, like the Atmel-ICE, SAM-ICE or SAM D5x/E5x Xplained Pro (SAM D5x/E5x evaluation kit) embedded debugger.

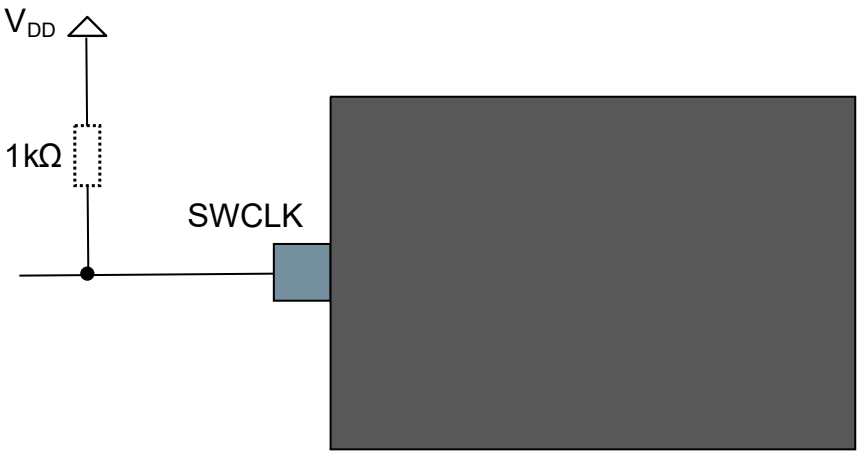
Refer to the Atmel-ICE, SAM-ICE or SAM D5x/E5x Xplained Pro user guides for details on debugging and programming connections and options. For connecting to any other programming or debugging tool, refer to that specific programmer or debugger's user guide.

The SAM D5x/E5x Xplained Pro evaluation board supports programming and debugging through the onboard embedded debugger, hence no external programmer or debugger is needed.

**Important:** A pull-up resistor on the SWCLK pin is critical for reliable operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 60-13.** SWCLK Circuit Connections




**Note:** JTAG and Serial Wire Interface recommendations can be found in the *"Basic 32-Bit MCU Design and Troubleshooting Checklist"* guide, available on the Microchip web site, www.microchip.com/ .

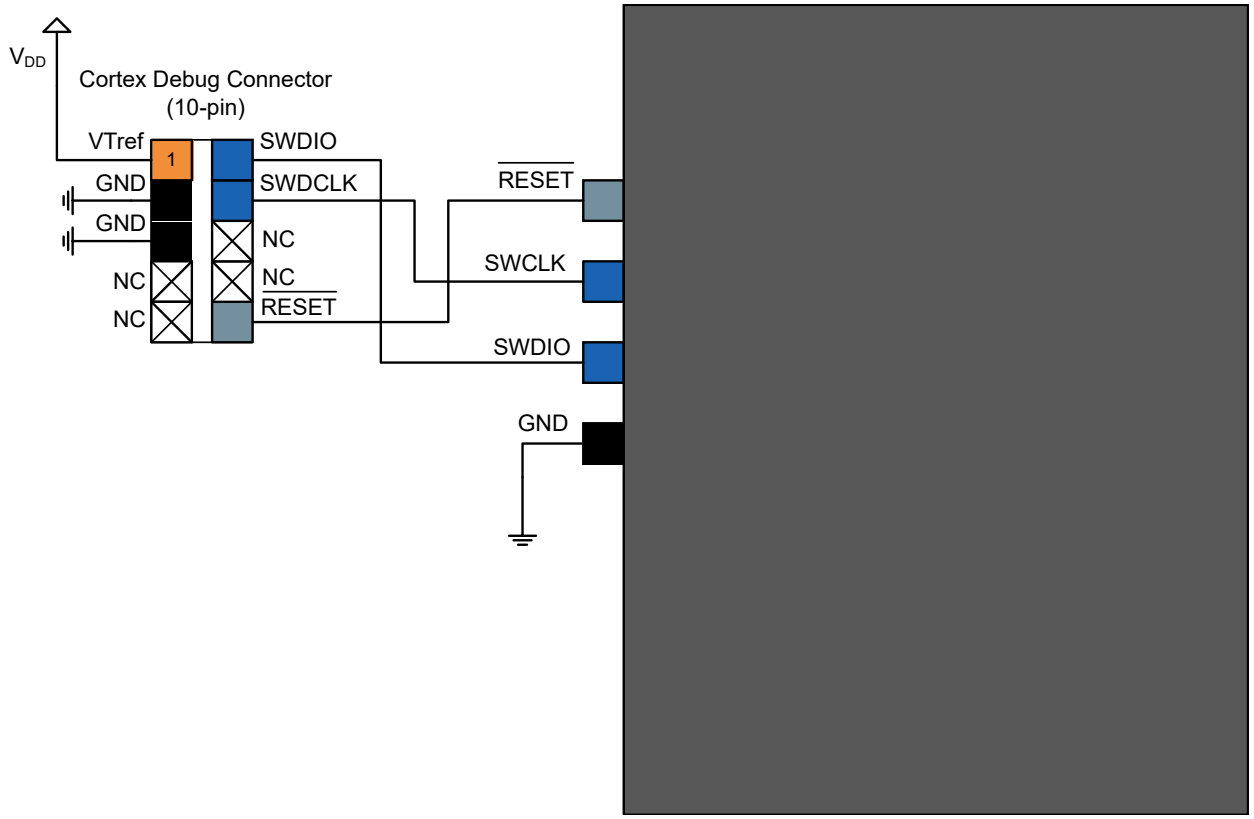
**Table 60-6.** SWCLK Circuit Connections

| Pin Name | Description | Recommended Pin Connection |
|---|---|---|
| SWCLK | Serial wire clock pin | Pull-up resistor 1 kΩ |

### 60.7.1. Cortex Debug Connector (10-pin)

For debuggers and/or programmers that support the Cortex Debug Connector (10-pin) interface the signals should be connected as shown in Figure 60-14 with details described in Table 60-7.

**Figure 60-14.** Cortex Debug Connector (10-pin)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 60-7.** Cortex Debug Connector (10-pin)

| Header Signal Name | Description |
|---|---|
| SWDCLK | Serial wire clock pin |
| SWDIO | Serial wire bidirectional data pin |
| $\overline{\text{RESET}}$ | Target device reset pin, active low |
| VTref | Target voltage sense, should be connected to the device $V_{DD}$ |
| GND | Ground |

## 60.7.2. 20-pin IDC JTAG Connector

For debuggers and/or programmers that support the 20-pin IDC JTAG Connector, e.g. the SAM-ICE, the signals should be connected as shown in Figure 60-15 with details described in Table 60-8.

**Figure 60-15.** 20-pin IDC JTAG Connector

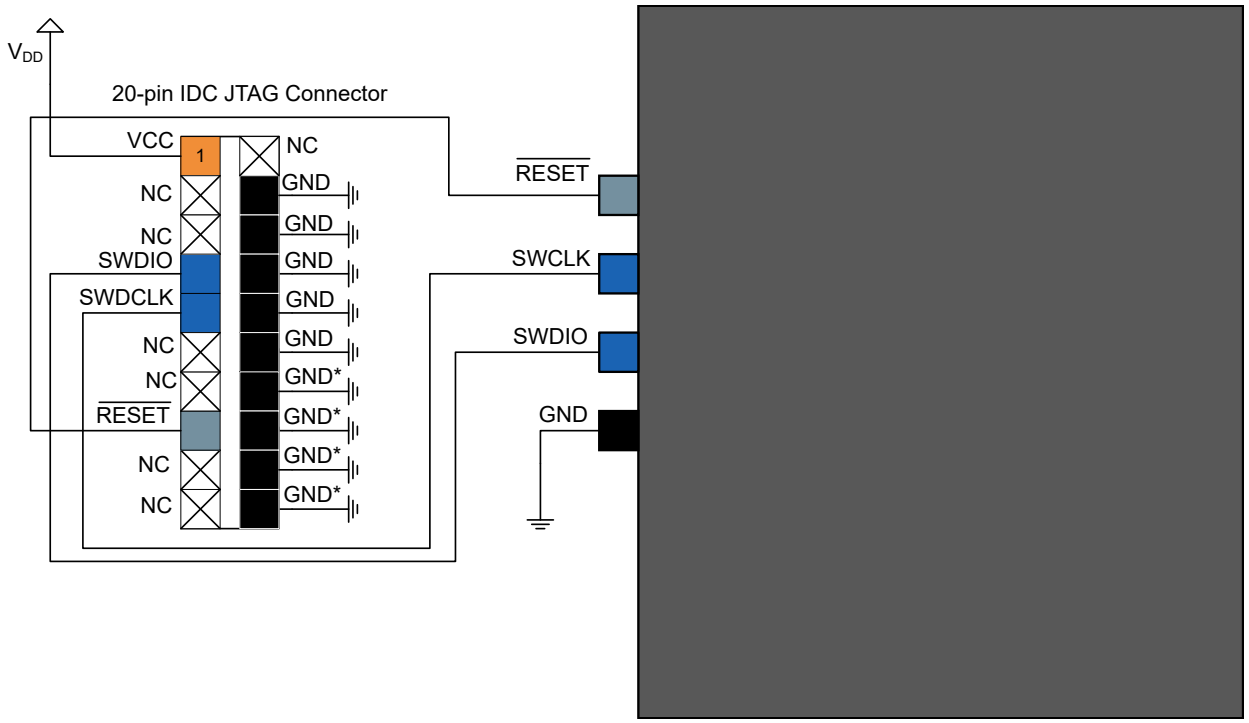



**Table 60-8.** 20-pin IDC JTAG Connector

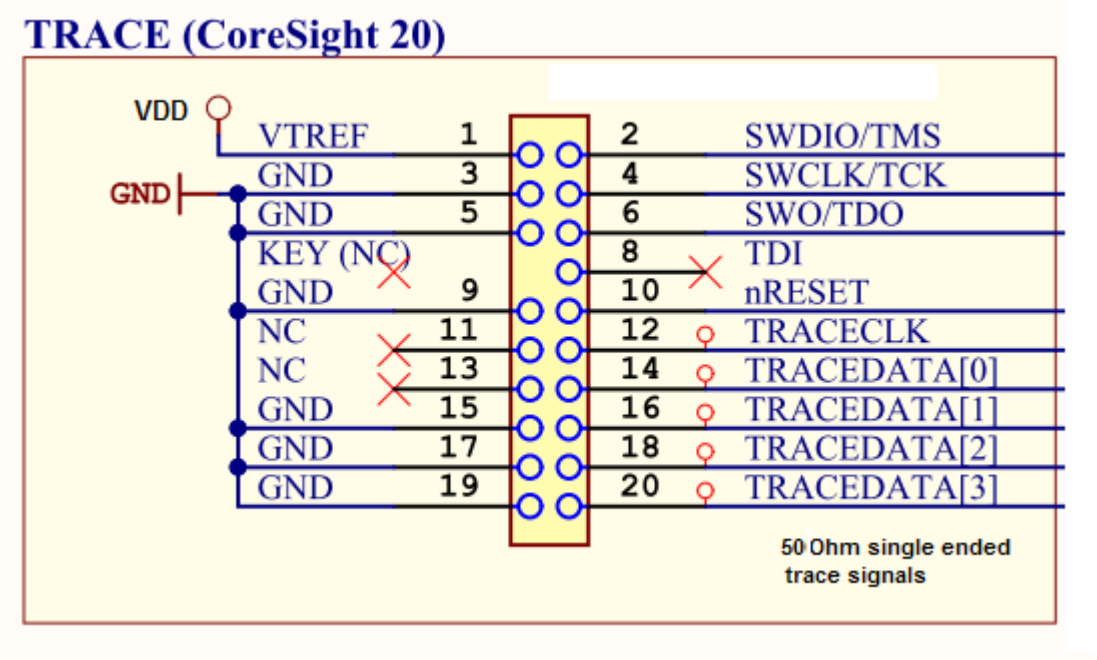
| Header Signal Name | Description |
|---|---|
| SWDCLK | Serial wire clock pin |
| SWDIO | Serial wire bidirectional data pin |
| $\overline{\text{RESET}}$ | Target device reset pin, active low |
| VCC | Target voltage sense, should be connected to the device $V_{DD}$ |
| GND | Ground |
| GND* | These pins are reserved for firmware extension purposes. They can be left unconnected or connected to GND in normal debug environment. They are not essential for SWD in general. |

## 60.7.3. Trace (CoreSight 20) Connector

The Trace Port Interface Unit (TPIU) takes data from the Embedded Trace Module (ETM) and allows debugger communication to ETM. The following figure shows the connection diagram.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

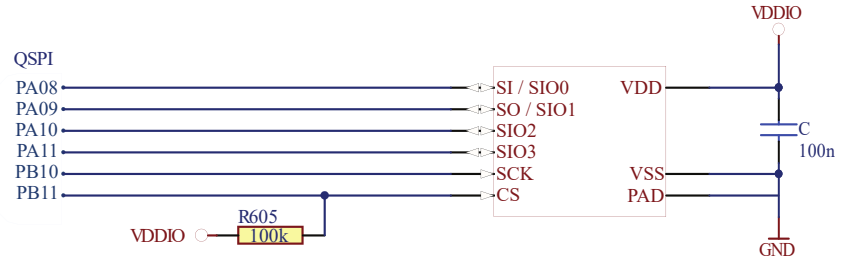
**Figure 60-16.** Trace (CoreSight 20) Connector Diagram




## 60.8. QSPI Interface

**Table 60-9.** QSPI Interface Pins and Connections

| Pin Name | Recommended Pin Connection | Description |
|---|---|---|
| PA08 - PA11 | Application dependent | QSPI I/O lines |
| PB10 | Application dependent | QSPI Clock |
| PB11 | Application dependent | QSPI Chip Select |




**Note:** Signal integrity can be improved by adding series resistors on each QSPI line. The resistor value should be based on the corresponding I/O pin drive strength (decided by PINCFGn.DRVSTR) and PCB trace impedance. It is recommended to do simulation using the device IBIS files to choose the correct termination and PCB trace impedance combination.

## 60.9. USB Interface

The USB interface consists of a differential data pair (D+/D-) and a power supply (VBUS, GND). Refer to the Electrical Characteristics section for operating voltages which will allow USB operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

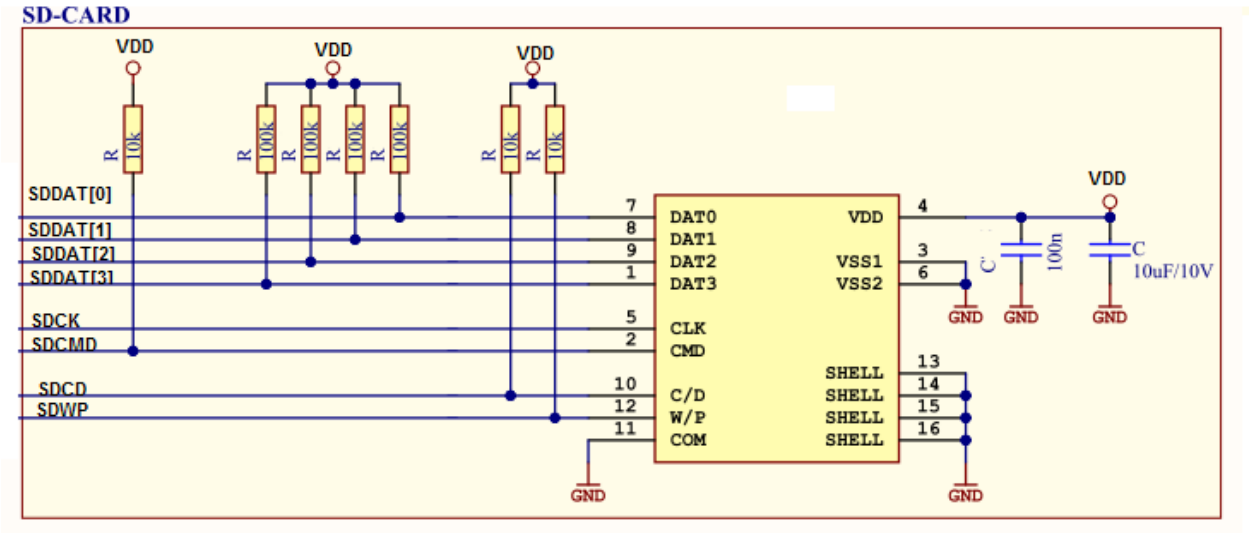
**Table 60-10.** USB Interface Checklist

| Signal Name | Recommended Pin Connection | Description |
|---|---|---|
| D+ | • The impedance of the pair must be matched on the PCB to minimize reflections.<br>• USB differential tracks must be routed with the same characteristics (length, width, number of vias, etc.)<br>• For a tightly coupled differential pair, the signal routing must be as parallel as possible, with a minimum number of angles and vias. | USB full speed/low speed positive data upstream pin |
| D- | | USB full speed/low speed negative data upstream pin |

USB Host and Device Interface recommendations can be found in the *"Basic 32-Bit MCU Design and Troubleshooting Checklist"* guide, available on the Microchip web site, www.microchip.com/.

## 60.10. SDHC Interface

The SD/MMC Host Controller (SDHC) is compliant with the SD Host Controller Standard specifications. There are two instances of SDHC available on this device: SDHC0 and SDHC1. The typical connection diagram is shown in the following figure.





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 61. Conventions

## 61.1. Numerical Notation

**Table 61-1.** Numerical Notation

| Symbol | Description |
|---|---|
| 165 | Decimal number |
| 0b0101 | Binary number |
| '0101' | Binary numbers are given without prefix if unambiguous |
| 0x3B24 | Hexadecimal number |
| X | Represents an unknown or do not care value |
| Z | Represents a high-impedance (floating) state for either a signal or a bus |

## 61.2. Memory Size and Type

**Table 61-2.** Memory Size and Bit Rate

| Symbol | Description |
|---|---|
| KB (kbyte) | kilobyte ($2^{10}$ = 1024) |
| MB (Mbyte) | megabyte ($2^{20}$ = 1024*1024) |
| GB (Gbyte) | gigabyte ($2^{30}$ = 1024*1024*1024) |
| b | bit (binary '0' or '1') |
| B | byte (8 bits) |
| 1kbit/s | 1,000 bit/s rate (not 1,024 bit/s) |
| 1Mbit/s | 1,000,000 bit/s rate |
| 1Gbit/s | 1,000,000,000 bit/s rate |
| word | 32 bit |
| half-word | 16 bit |

## 61.3. Frequency and Time

**Table 61-3.** Frequency and Time

| Symbol | Description |
|---|---|
| kHz | 1 kHz = $10^3$ Hz = 1,000 Hz |
| MHz | 1 MHz = $10^6$ Hz = 1,000,000 Hz |
| GHz | 1 GHz = $10^9$ Hz = 1,000,000,000 Hz |
| ms | 1 ms = $10^{-3}$s = 0.001s |
| µs | 1 µs = $10^{-6}$s = 0.000001s |
| ns | 1 ns = $10^{-9}$s = 0.000000001s |

## 61.4. Registers and Bits

**Table 61-4.** Register and Bit Mnemonics

| Symbol | Description |
|---|---|
| R/W | Read/Write accessible register bit. The user can read from and write to this bit. |
| R | Read-only accessible register bit. The user can only read this bit. Writes will be ignored. |
| W | Write-only accessible register bit. The user can only write this bit. Reading this bit will return an undefined value. |
| BIT | Bit names are shown in uppercase. (Example ENABLE) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 61-4.** Register and Bit Mnemonics (continued)

| Symbol | Description |
|---|---|
| FIELD[n:m] | A set of bits from bit n down to m. (Example: PINA[3:0] = {PINA3, PINA2, PINA1, PINA0} |
| Reserved | Reserved bits are unused and reserved for future use. For compatibility with future devices, always write reserved bits to zero when the register is written. Reserved bits will always return zero when read.<br>Reserved bit field values must not be written to a bit field. A reserved value will not be read from a read-only bit field. |
| PERIPHERAL*i* | If several instances of a peripheral exist, the peripheral name is followed by a number to indicate the number of the instance in the range 0-n. PERIPHERAL0 denotes one specific instance. |
| Reset | Value of a register after a power Reset. This is also the value of registers in a peripheral after performing a software Reset of the peripheral, except for the Debug Control registers. |
| SET/CLR | Registers with SET/CLR suffix allows the user to clear and set bits in a register without doing a read-modify-write operation. These registers always come in pairs. Writing a '1' to a bit in the CLR register will clear the corresponding bit in both registers, while writing a '1' to a bit in the SET register will set the corresponding bit in both registers. Both registers will return the same value when read. If both registers are written simultaneously, the write to the CLR register will take precedence. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 62. Acronyms and Abbreviations

The below table contains acronyms and abbreviations used in this document.

**Table 62-1.** Acronyms and Abbreviations

| Abbreviation | Description |
|---|---|
| AC | Analog Comparator |
| ADC | Analog-to-Digital Converter |
| ADDR | Address |
| AES | Advanced Encryption Standard |
| AHB | Advanced High-performance Bus |
| AMBA | Advanced Microcontroller Bus Architecture |
| APB | AMBA Advanced Peripheral Bus |
| AREF | Analog Reference Voltage |
| BOD | Brown-out Detector |
| CAL | Calibration |
| CC | Compare/Capture |
| CCL | Configurable Custom Logic |
| CLK | Clock |
| CRC | Cyclic Redundancy Check |
| CTRL | Control |
| DAC | Digital-to-Analog Converter |
| DAP | Debug Access Port |
| DFLL | Digital Frequency Locked Loop |
| DPLL | Digital Phase Locked Loop |
| DMAC | DMA (Direct Memory Access) Controller |
| DSU | Device Service Unit |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| EIC | External Interrupt Controller |
| EVSYS | Event System |
| FDPLL | Fractional Digital Phase Locked Loop, also DPLL |
| FREQM | Frequency Meter |
| GCLK | Generic Clock Controller |
| GMII | Gigabit Media Independent Interface |
| GND | Ground |
| GPIO | General Purpose Input/Output |
| $I^2C$ | Inter-Integrated Circuit |
| IF | Interrupt Flag |
| INT | Interrupt |
| MBIST | Memory Built-In Self-Test |
| MEM-AP | Memory Access Port |
| MIB | Management Information Base |
| MII | Media Independent Interface |
| MTB | Micro Trace Buffer |
| NMI | Non-maskable Interrupt |
| NVIC | Nested Vector Interrupt Controller |
| NVM | Nonvolatile Memory |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

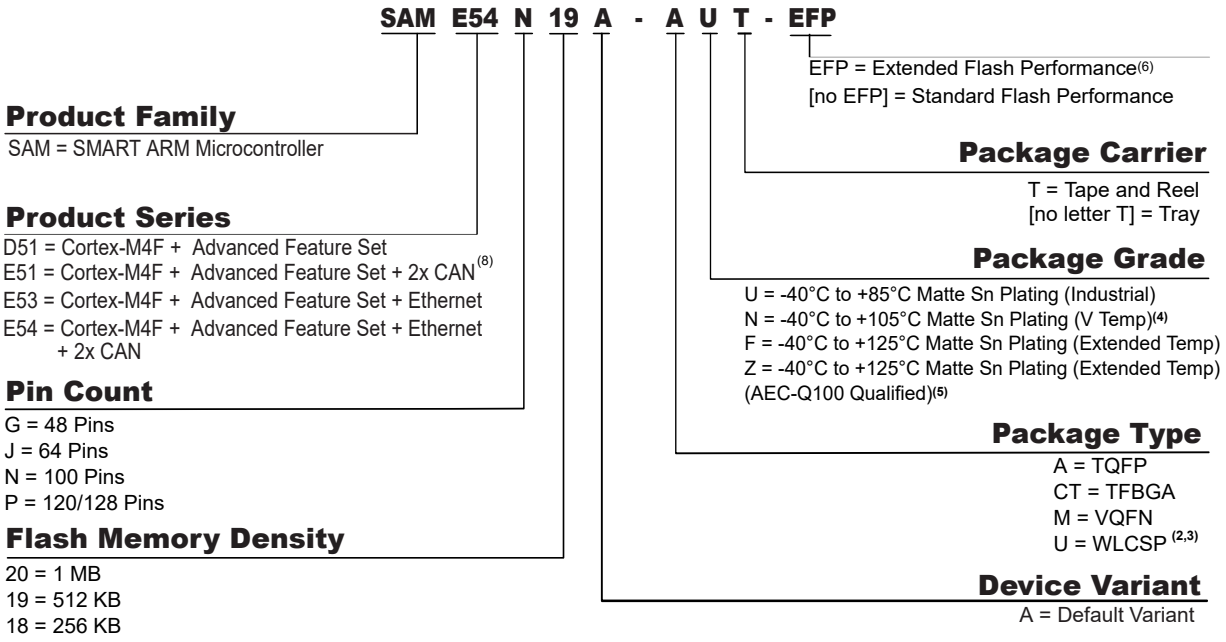
**Table 62-1.** Acronyms and Abbreviations (continued)

| Abbreviation | Description |
|---|---|
| NVMCTRL | Nonvolatile Memory Controller |
| OSC | Oscillator |
| PAC | Peripheral Access Controller |
| PC | Program Counter |
| PER | Period |
| PM | Power Manager |
| POR | Power-on Reset |
| PORT | I/O Pin Controller |
| PTC | Peripheral Touch Controller |
| PWM | Pulse-Width Modulation |
| RAM | Random-Access Memory |
| REF | Reference |
| RMII | Reduced Media Independent Interface |
| RTC | Real-Time Counter |
| RX | Receiver/Receive |
| SEES | SmartEEPROM Sector |
| SEEP | SmartEEPROM Page |
| SERCOM | Serial Communication Interface |
| SMBus | System Management Bus |
| SNAP | Sub-Network Access Protocol |
| SP | Stack Pointer |
| SPI | Serial Peripheral Interface |
| SRAM | Static Random Access Memory |
| SUPC | Supply Controller |
| SWD | Serial Wire Debug |
| TC | Timer/Counter |
| TRNG | True Random Number Generator |
| TX | Transmitter/Transmit |
| ULP | Ultra Low-Power |
| USART | Universal Synchronous and Asynchronous Serial Receiver and Transmitter |
| USB | Universal Serial Bus |
| $V_{DD}$ | Common voltage to be applied to VDDIO and VDDANA |
| $V_{DDIO}$ | Digital Supply Voltage |
| $V_{DDANA}$ | Analog Supply Voltage |
| VREF | Voltage Reference |
| WDT | Watchdog Timer |
| XOSC | Crystal Oscillator |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 63. Product Identification System

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.





© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

# 64. Revision History

**Table 64-1.** Revision N. - 03/2026

| Section Name or Type | Change Description |
|---|---|
| Configuration Summary | • Updated Table 1-2 with new data memory specifications |
| CMCC | • Updated the LCKWAY bitfield with the addition of a bitfield value table in the LCKWAY Register<br>• Updated the bitfield value table for the WAY bitfield for the MAINT1 Register |
| Clock System | • Added a new note to the *Overview* for *Register Synchronization* |
| RAMECC | • Defined the disabling of the ECC in *Principle of Operation* |
| PM | • Defined wakeup from RTC Tamper in *Sleep Modes* |
| WDT | • Updated the Signal name in the EWOFFSET bitfield of the EWCTRL Register |
| DMAC | • Updated the Equations in *Addressing* |
| NVMCTRL | • Added a new line item to *Procedure for Manual Page Writes (WMODE=MAN)* in *NVM Write*<br>• Added a new note to *SmartEEPROM Sector Reallocation* |
| OSCCTRL | • Redefined line item one under *Closed-Loop Operation* in *Basic Operation* to explain setting GCLK.PCHCTRL0.GEN |
| PORT | • Updated the PINCFG Register with a corrected Register Property, and added a note related to the reset value for the DRVSTR bitfield. |
| SERCOM USART | • Updated the notes for the RXINV and TXINV bitfields in the CTRLA Register<br>• Updated the definition for CTRLA.FORM in the PMODE bitfield of the CTRLB Register<br>• Updated the definition for CTRLA.MODE in the MAXITER bitfield of the CTRLC Register |
| SERCOM $I^2C$ | • Updated the equation in the SDASETUP bitfield of the CTRLC Register |
| CAN | • Updated the Signal names in<br>– *Test Modes*<br>– *Sleep Mode Operation*<br>– *Extended Message ID Filter Element*<br>• Minor updates to signal names or other Register properties were made in the following registers:<br>– CCCR<br>– NBTP |
| SD/MMC | • Added text related to the GCLK Core Clock to the BASECLK bitfield of the CA0R Register |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-1.** Revision N. - 03/2026 (continued)

| Section Name or Type | Change Description |
|---|---|
| CCL | • Removed erroneous content under *Timer/Counter Inputs (TC)* in *Truth Table Inputs Selection*<br>• Removed erroneous content from the bitfield values table for the INSELx bitfield in the LUTCTRL Register |
| AES | • Added a new paragraph to *Galois Counter Mode (GCM)* |
| ADC | • Added a new note to *Host - Client Operation* |
| AC | • Added a new note to *Offset Compensation* |
| TC | • Updated the text of COUNT32 under *Counter Mode*<br>• Updated the CTRLBSET with a new Register Property |
| TCC | • Updated the diagram in *Circular Buffer* with a new signal name<br>• Updated the WAVE Register with a new Register Property |
| Electrical Characteristics at 85°C | • Removed erroneous information from table 54-59 and removed the diagram |

**Table 64-2.** Rev M. - 12/2024

| Section Name or Type | Change Description |
|---|---|
| Features | • Added content for Related Devices to the Features set |

**Table 64-3.** Rev L. - 08/2023

| Section Name or Type | Change Description |
|---|---|
| General | No content changes compared with the previous revision. |

**Table 64-4.** Rev K. - 03/2023

| Section Name or Type | Change Description |
|---|---|
| Configuration Summary | • Updated Memory columns with new titles and specifications |
| Block Diagram | • Updated the Block Diagram with a new figure |
| High-Speed Bus System | • Updated *Configuration* with a naming clarification in Table 10-2 |
| DSU | • Updated the STATUSB register with new verbiage for the PROT bit |
| DMAC | • Added hyphens to clarify information in the tables for the TRIGACT and TRIGSRC bitfields in the CHCTRLA Register |
| NVMCTRL | • Updated the verbiage in *Memory Organization*<br>• Restructured the line items and added new information to *Security Bit Set Procedure*<br>• Updated bit naming in *Chip Erase*<br>• Updated commands in the table for the CMD bitfield in the CTRLB Register<br>• Reorganized the table for the PSZ bitfield in the PARAM register<br>• Updated the bitfield reset properties for the PSZ and SBLK bitfields in the SEESTAT register |
| SERCOM USART | • Clarified verbiage for the Tx buffer in *Data Transmission*<br>• Added a new note to *LIN Host*<br>• Added a note to the HDRDLY and BRKLEN bitfields in the CTRLC register |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 64-4.** Rev K. - 03/2023 (continued)

| Section Name or Type | Change Description |
|---|---|
| USB | • Updated the FNUM register, and removed bits 2-0 MFNUM.<br>• Updated the CTRLB register and removed these bits: 8,7,6,5<br>• Updated the INTENCLR register and removed Bit 1 MSOF<br>• Updated the INTENSET register and removed Bit 1 MSOF<br>• Updated the INTFLAG register and removed Bit 1 MSOF<br>• Updated the EPCFGn register and removed bit 7 NYETDIS<br>• Updated the CTRLB register and removed bits TSTK and TSTJ<br>• Updated the HSOFC register and removed one table under the FLENCE bit<br>• Updated the STATUS register bit SPEED<br>• Updated the FNUM register, removed the Bit 2-0 MFNUM<br>• Updated the FLENHIGH register bit 7:0 FLENHIGH<br>• Updated the BINTERVALn register Bits 7:0 – BINTERVAL |
| CAN | • Updated the NBTP register with new verbiage |
| ADC | • Added a new section Rail-to-Rail Operation |
| TC | • Added line item 4 to Capture Operations |
| PDEC | • Clarified signal names in the Block Diagram |
| Electrical Characteristics at 85°C | • Added a row for Tstart to Table 54-44 in Crystal Oscillator (XOSC) Characteristics |
| Electrical Characteristics at 105°C | • Added a new table to Crystal Oscillator (XOSC) Characteristics |
| Electrical Characteristics at 125°C | • Added a new table to Crystal Oscillator (XOSC) Characteristics |
| Schematic Checklist | • Added and clarified verbiage in the Introduction<br>• Added a new Important note to *Operation in Noisy Environment*<br>• Added a new note and removed obsolete content from External Reset Circuit<br>• Added a new note and clarified verbiage in Programming and Debug Ports<br>• Removed obsolete information from USB Interface |

### Rev J. - 08/2022

Terminology used in this document may not match with the contents of the current revision of the device errata or other Microchip documentation and collateral. If there are any questions or concerns regarding terminology, contact a Microchip Support or Sales Representative.

The following updates were performed in this revision:

**Table 64-5.** Rev J. - 08/2022

| Section Name or Type | Change Description |
|---|---|
| GENERAL | • The $I^2C$, SPI and $I^2S$ standards use the terminology "Master" and "Slave". The equivalent Microchip terminology used in this document is "Host" and "Client" respectively. |
| GCLK | • Added a new step to Selecting the Clock Source for a Peripheral |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-5.** Rev J. - 08/2022 (continued)

| Section Name or Type | Change Description |
|---|---|
| RTC | • Updated the following registers with new register properties:<br>– COUNT in COUNT32 Mode<br>– COUNT in COUNT16 Mode |
| OSCCTRL | • Updated Additional Features with new notes and text for changing the reference clock source |
| EVSYS | • Updated the naming of the CHANNELn Register |
| SERCOM USART | • Updated Start-of-Frame Detection with new content for the first paragraph |
| SERCOM SPI | • Added new text to I/O Lines |
| AC | • Added third note to *Window Operation*<br>• Added a note to the ENABLE bit in the CTRLA Register<br>• Updated the naming for the COMPCTRLn Register |
| TC | • Removed an erroneous reference to the DMAOS bitfield in the CTRLA Register for 8-bit Mode, 16-bit Mode, and 32-bit Mode<br>• Replaced the bitfield description for the CMD bitfield in the CTRLBCLR Register for 8-bit Mode, 16-bit Mode, and 32-bit Mode<br>• Added an Important note to the CMD bitfield in the CTRLBSET Register for 8-bit Mode, 16-bit Mode, and 32-bit Mode<br>• Updated the Register Property for the following Registers:<br>– COUNT 8-bit Mode<br>– CCx 8-bit Mode<br>– COUNT 16-bit Mode<br>– COUNT 32-bit Mode |
| TCC | • Added a new note to Counter Operation<br>• Added a note to the SWRST Bitfield in the CTRLA Register<br>• The Bitfield description for the CMD bitfield was rewritten in the CTRLBCLR Register<br>• Added notes to the CMD and DIR Bitfields in the CTRLBSET Register<br>• Updated the Register properties for the following Registers:<br>– STATUS<br>– WAVE<br>– CCx<br>– PATTBUF<br>– PERBUF<br>– CCBUF<br>– COUNT |
| PTC | • The Functional Description was reworked to include references to Microchip Touch |
| $I^2S$ | • Added a note to Principle of Operation<br>• Added a note to the FSWIDTH bitfield of the CLKCTRLn Register |
| USB | • Restructured all Register sets and corrected register offsets |
| PDEC | • Updated the signal equation in Input Selection and Filtering |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-5.** Rev J. - 08/2022 (continued)

| Section Name or Type | Change Description |
|---|---|
| Electrical Characteristics at 85°C | • Updated the Analog Comparator Characteristics table with new Conditions values for Vhys, min, and max values<br>• Updated Table 54-13. I/O Pins Maximum Output Current with new values |
| Electrical Characteristics at 105°C | • Updated the Analog Comparator Characteristics table with new min and max values |
| Electrical Characteristics at 125°C | • Updated the Analog Comparator Characteristics table with new min and max values |

**Table 64-6.** Rev H. - 02/2022

| Section Name or Type | Change Description |
|---|---|
| Analog Comparators - AC | • Updated the Features with a new DAC0 input<br>• Updated the Block Diagram to display DAC0 instead of DAC<br>• Removed erroneous 0x3 information from the COMPCTRLx.SPEED entry in Comparator Configuration<br>• Updated the following registers:<br>– STATUSA with new info for the WSTATE0 bitfield<br>– COMPCTRLn with new information for the HYST Bitfield |
| Electrical Characteristics at 85°C | Updated the Analog Comparator Characteristics table in Analog Comparator (AC) Characteristics. Added a new note to the table. |
| Electrical Characteristics at 105°C | Updated the Analog Comparator Characteristics table in Analog Comparator (AC) Characteristics. Added a new note to the table. |
| Electrical Characteristics at 125°C | Updated the Analog Comparator Characteristics table in Analog Comparator (AC) Characteristics. Added a new note to the table. |

**Table 64-7.** Rev G. - 01/2021

| Section Name or Type | Change Description |
|---|---|
| General | Along with the updates listed below, numerous typographical and minor editorial updates were made to this document. |
| Features | Updated Memories to show proper nomenclature for SmartEEPROM. |
| Configuration Summary | Updated Table 1-2 with new $I^2S$ specs for SAME51G18 and SAME51G19. |
| I/O Multiplexing and Considerations | Updated Table 6-34 GPIO Clusters, removed erroneous GPIO designations for the 100 pin package. |
| Memories | Updated Table 9-4 for bit position 75:64. |
| CMCC | Removed erroneous bitfields from the TYPE Register. |
| DSU | • Updated SmartEEPROM nomenclature in Chip Erase<br>• Updated references to SmartEEPROM in Table 12-3 of 32-bit Cyclic Redundancy Check CRC32<br>• Updates table 12-6 in System Services Availability when Accessed Externally and Device is Protected with proper verbiage for SmartEEPROM<br>• Updated ENTRY0 and ENTRY1 registers with correct numbering |
| Clock System | Updated the Clock Request Routing Diagram in On Demand Clock Requests. |
| GCLK | Updated the GENCTRLn register with a new bitfield length for the SRC bit, and added new data to the end of table 14-4 Generator Clock Source Selection. |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 64-7.** Rev G. - 01/2021 (continued)

| Section Name or Type | Change Description |
|---|---|
| MCLK | • Updated the table in Peripheral Clock Masking with new values for the ADC, NVMCTRL, and removed a reference to the PTC<br>• Updated the bitfield numbering and reset values in the following registers:<br>– AHBMASK<br>– APBCMASK |
| PM | • Updated figure 18-2 in Sleepwalking<br>• Updated the register offset for the PWSAKDLY Register |
| SUPC | Removed an erroneous BOD12 reference from Enabling, Disabling, and Resetting. |
| RTC | Updated the COUNT32 Register with a new note. |
| DMAC | Updated the following registers with new bitfield verbiage:<br>• BASEADDR<br>• WRBADDR<br>• SRCADDR<br>• DSTADDR |
| GMAC | • Updated the following registers with new bitfield information for bit 27:<br>– ISR<br>– IER<br>– IDR<br>– IMR<br>• Updated the TPB1ADR bit in the TPSF Register<br>• Updated the RPB1ADR bit in the RPSF Register<br>• Added the TPFCP Register<br>• Renamed the bitfield in the TLPITI Register from RLPITI to TLPITI |
| NVMCTRL | • Updated verbiage for SmartEEPROM throughout the entire chapter<br>• Corrected typographical errors in Safe Flash Update Using Dual Banks<br>• Updated the CELCK Description in the table for the CMD bit in the CTRLB Register |
| ICM | Updated the Functional Description Overview topic with new verbiage for the end of list marker and WRAP bit setting. |
| OSCCTRL | Updated references for the SWBCK bit to the SWBEN bit in the Clock Switch section of Clock Failure Detection Operation. |
| OSC32KCTRL | • Added a new note to 32 kHz Ultra Low-Power Internal Oscillator (OSCULP32K) Operation, and removed erroneous text references to the CALIB bit<br>• Updated the OSCULP32K Register with the removal of the erroneous CALIB bit, and converted it to a Reserved bit |
| EVSYS | • Updated the following registers with new bitfield numbering:<br>– CHANNEL<br>– USERm |
| PORT | Updated the bitfield numbering for the PORTEIx, EVACTx, and PIDx bits in the EVCTRL Register. |
| SERCOM $I^2C$ | • Corrected references for the LOWTOUTEN bit in Initialization<br>• Updated the CTRLA (Slave) Register with corrections to the name for the LOWTOUTEN bit<br>• Added a new LENERR bit to the STATUS (Slave) Register<br>• Updated the CTRLA (Master) Register with corrections to the name for the LOWTOUTEN bit |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-7.** Rev G. - 01/2021 (continued)

| Section Name or Type | Change Description |
|---|---|
| QSPI | • Updated Serial Clock Phase and Polarity with new values for the Capture SCK Edge column in table 37-2<br>• Added a new note to QSPI Serial Memory Mode<br>• Updated the bitfield Access for the ENABLE bit in the CTRLA Register |
| USB | The STATUS (Device Registers - Common) was updated with new bit field data. |
| CAN | Updates were made to the following Registers for bitfield naming, numbering, and typographical corrections:<br>• MRCFG<br>• ILE<br>• NDAT1<br>• NDAT2<br>• TXBRP<br>• TXBAR<br>• TXBCR<br>• TXBTO<br>• TXBCF<br>• TXBTIE<br>• TXBCIE<br>• TXEFS |
| SDHC | • Removed erroneous Asynchronicity text in Clocks<br>• Updated the MAXBLKL bitfield with proper numbering in the CA0R Register<br>• Updated the PVR Register with the addition of the DRVSEL bitfield |
| CCL | • Updated the Block Diagram to properly display only one layer<br>• Updated Signal Description with new Pin names<br>• Updated the LUTCTRL Register name |
| ADC | Removed erroneous RBSSW bitfield from the SYNCBUSY Register. |
| TC | Updated the Note in Capture Operations. |
| TCC | • Updated the Block Diagram to display six Compare/Capture units<br>• Updated Principle of Operation to state six compare/Capture channels instead of four.<br>• Updated the following Registers:<br>– CTRLA with a new bit numbering for the CPTENx bitfield<br>– EVCTRL with a new bit numbering for the MCEOx and MCEIx bitfields<br>– The following registers have updates made to the MCx bitfields:<br>• INTENCLR<br>• INTENSET<br>• INTFLAG<br>– STATUS with a new bitfield numbering for CMPx and CCBUFVx |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-7.** Rev G. - 01/2021 (continued)

| Section Name or Type | Change Description |
|---|---|
| PCC | • Removed erroneous text for transfer status signals from Principle of Operation<br>• Updated the text for line item number five in With DMAC<br>• Updated the following Registers with the removal of erroneous bitfields:<br>– IER<br>– IDR<br>– IMR<br>– ISR |
| PDEC | Added a new section to the chapter titled Secure Decoder Detection. |
| Electrical Characteristics at 85°C | • Added a new note to I/O Pin Characteristics<br>• Removed erroneous BIASREFBUF references from the Power Consumption table in Analog-to-Digital Converter (ADC) Characteristics<br>• Removed the BANDGAP reference from Table 54-34 in Voltage References<br>• In Internal Ultra Low Power 32 kHz RC Oscillator (OSCULP32K) Characteristics an erroneous Calibration column was removed from table 54-49 |
| Appendix A | Added a new Appendix section for ISELED Specifications. |

**Table 64-8.** Rev. F - 05/2020

| Section Name or Type | Change Description |
|---|---|
| Features | • Updated the number of Temperature sensors<br>• Corrected Wake-up pins from 4 to 5 |
| I/O Multiplexing and Considerations | Added a new note to Table 6-34 GPIO Clusters. |
| Power Supply and Startup Considerations | • Updated VDDIO and VDDANA information in Power Supplies<br>• Updated the Power Domain Block Diagram to properly display the POR<br>• Updated Section 7.3.3 I/O Pins with a more detailed explanation of the pins |
| GCLK | • Corrected the number of Peripheral Channels from 64 to 48 in Principle of Operation<br>• Removed CTRLA register from section 15.8 Register Description |
| MCLK | Added a new note to the AHBMASK register, and modified the access attribute of bits 11,7 and 5 from "R" to "R/W." |
| DMAC | • Added disable information to the SWRST bit in the CTRL register<br>• Updated the PRICTRL0 Register with new bit information for the QOS bit |
| RTC | • Corrected the Tamper channels from 4 to 5 in Tamper Detection<br>• Added a new note to Timestamp |
| EVSYS | Updated the Register offset equations for the following registers:<br>• CHINTENCLR<br>• CHINTENSET<br>• CHINTFLAG<br>• CHSTATUSn |
| SERCOM USART | Corrected the Baud Rate Divider in the Clock Generation figure for Clock Generation and Selection. |
| $I^2S$ | Updated Figure 51-8 and 51-9 in $I^2S$ Application Examples. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-8.** Rev. F - 05/2020 (continued)

| Section Name or Type | Change Description |
|---|---|
| USB | • Updated the EPSTATUSn register with new table information for the BK1RDY bit<br>• Removed an incorrect reference to INTFLAGA in Figure 38.9 Device Interrupt<br>• Removed the MFNUM bit from the ENUM register |
| CAN | • Updated Timestamp Generation with a revised bit name for CCCR.BRSE and a note about bit rate switching<br>• Corrected the TSC value from 15 to 16 in Timeout Counter |
| TRNG | Updated reset information for the DATA register. |
| TC | Corrected GCLK_TC to GCLK_TCx and CLK_TC_CNT to CLK_TCx_CNT throughout the chapter. |
| TCC | • Corrected GCLK_TCC to GCLK_TCCx throughout the chapter<br>• Corrected the Register representation for:<br>– Pattern<br>– Pattern Buffer |
| ADC | Updated the REFSEL bit table in the REFCTRL register. |
| AES | The name of the register changed from DATA to INDATA |
| DAC | Updated the CTRLB register with new information for the REFSEL bit. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-8.** Rev. F - 05/2020 (continued)

| Section Name or Type | Change Description |
|---|---|
| Electrical Characteristics at 85°C | • Updated the note for the Absolute Maximum Ratings table<br>• Updated the table for General Operating Ratings<br>• Updated the note in Injection Current<br>• Updated Supply Characteristics with a new table<br>• Updated the tables in I/O Pin Characteristics with correct conditions and notes<br>• Replaced the tables and associated notes with new information in Brown-Out Detectors (BOD) Characteristics<br>• Replaced the tables with new information in Analog-to-Digital Converter (ADC) Characteristics<br>• Updated the tables in Analog Comparator (AC) Characteristics with correct parameter name<br>• Updated the tables NVM Characteristics with correct format, parameter name, and new notes<br>• Updated Crystal Oscillator (XOSC) Characteristics with a simplified schematic for Oscillator Connection<br>• Updated External 32 kHz Crystal Oscillator (XOSC32K) Characteristics with a simplified schematic for Oscillator Crystal Connection and corrected the Load Capacitance equation<br>• Removed 25°C data from Ultra-Low-Power Internal 32 kHz RC Oscillator Electrical Characteristics in Internal Ultra Low Power 32 kHz RC Oscillator (OSCULP32K) Characteristics<br>• Merged the current values in the DFLL48M Power Consumption table in Digital Frequency Locked Loop (DFLL48M) Characteristics<br>• Reorganized SERCOM in SPI Mode Timing to show the figures before the table<br>• Added $t_{MOH}$ parameter value in Table 54-56. SPI Timing Characteristics and Requirements<br>• Added max value for parameter $t_{SCKR}$ and $t_{SCKf}$ in Table 54-56. SPI Timing Characteristics and Requirements<br>• Added new note to QSPI Maximum Frequency Examples in QSPI Characteristics<br>• Reorganized GMAC Characteristics to display the diagrams before the tables<br>• Reorganized $I^2S$ Characteristics to display the images before the tables<br>• Updated Table 54-63. I2S Timing Characteristics and Requirements for parameter $f_{M_SCKO}$<br>• Added new figures and tables to PCC Characteristics<br>• Added the following new sections:<br>– TC Characteristics<br>– TCC Characteristics<br>– PDEC Characteristics |
| Electrical Characteristics at 105°C | • Added a new note to General Operating Ratings<br>• Added new tables to Brown-Out Detectors (BOD) Characteristics 105°C<br>• Added new tables and notes to Analog-to-Digital Converter (ADC) Characteristics (105°C)<br>• Removed the Multiple Crystal Oscillator Electrical Characteristics Table from Crystal Oscillator (XOSC) Characteristics (105°C)<br>• Updated NVM Characteristics with correct format, parameter name, and a new note |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-8.** Rev. F - 05/2020 (continued)

| Section Name or Type | Change Description |
|---|---|
| Electrical Characteristics at 125°C | • Updated the table in General Operating Ratings with a new note<br>• Updated Injection Current with new notes<br>• Updated Brown-Out Detectors (BOD) Characteristics (125°C) with new tables and notes<br>• Removed the Power Supply Current Requirement table<br>• Updated Analog-to-Digital Converter (ADC) Characteristics (125°C) with new tables and new notes<br>• Added a new note to the Analog Comparator Characteristics table for the operating voltage range.<br>• Updated NVM Characteristics (125C) with correct format, parameter name, and a new note<br>• Removed Multiple Crystal Oscillator Electrical Characteristics Table from Crystal Oscillator (XOSC) Characteristics (125°C)<br>• Removed 32 kHz Crystal Oscillator Electrical Characteristics table from External 32 kHz Crystal Oscillator (XOSC32K) Characteristics (125°C) |
| Package Information | • Updated Junction Temperature with a new table<br>• Added 120-ball TFBGA (AEC-Q100 Grade 1) |
| Schematic Checklist | • Updated the Schematics in Power Supply Connections with a 100nF connection |

**Table 64-9.** Revision E - 06/2019

| Section Name or Type | Change Description |
|---|---|
| Introduction | Updated sections:<br>• Updated Features<br>• Updated the Configuration Summary |
| Processor and Architecture | Updated Interrupt Line Mapping |
| CMCC | Updated sections:<br>• Data Cache Disable<br>• Instruction Cache Disable |
| GCLK | Updated the PCHCTRLm register |
| PM | Updated Backup Mode |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-9.** Revision E - 06/2019 (continued)

| Section Name or Type | Change Description |
|---|---|
| RTC | Count32 Registers updated:<br>• CTRLA<br>• EVCTRL<br>• INTENCLR<br>• INTENSET<br>• INTFLAG<br>• SYNCBUSY<br>• GPn<br>Count16 Registers updated:<br>• CTRLA<br>• EVCTRL<br>• INTENCLR<br>• INTENSET<br>• INTFLAG<br>• SYNCBUSY<br>• GPn<br>The following Clock registers were updated:<br>• CTRLA<br>• EVCTRL<br>• INTENCLR<br>• INTENSET<br>• INTFLAG<br>• SYNCBUSY<br>• GPn |
| DMAC | The following topics were updated:<br>• Initialization<br>• Enabling, Disabling and Resetting<br>• Transfer Descriptors<br>The following registers were updated:<br>• CTRL<br>• SWTRIGCTRL<br>• INTSTATUS<br>• BUSYCH |
| GMAC | The following topics were updated:<br>• Features<br>• Pause Frame Reception |
| OSCCTRL | The following topics were updated:<br>• Digital Phase Locked Loop (DPLL) Operation |
| SERCOM-SPI | Updated the following topics:<br>• Clock Generation |
| QSPI | Updated the following topics:<br>• Continuous Read Mode |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-9.** Revision E - 06/2019 (continued)

| Section Name or Type | Change Description |
|---|---|
| SDMMC | Updated the following registers:<br>• PSR<br>• PCR<br>• NISTR<br>• NISTER<br>• HC2R EMMC<br>• HC2R SDIO<br>• CA0R<br>• CA1R<br>• MCCAR<br>• ASAR<br>• PVRx<br>• MC1R<br>• ACR |
| ADC | Updated the INPUTCTRL register |
| AC | Updated the following sections:<br>• Signal Description |
| TC | Updated the following sections:<br>• Counter Mode |
| TCC | Updated the following sections:<br>• Capture Operations<br>Updated the following registers:<br>• WAVE |
| $I^2S$ | Updated the following sections:<br>• DMA Operation<br>Updated the following registers:<br>• CTRLA<br>• INTENCLR<br>• INTENSET<br>• INTFLAG<br>• SYNCBUSY<br>• TXCTRL<br>• RXCTRL |
| Electrical Characteristics at 85°C | Updated the following sections:<br>• General Operating Ratings<br>• Injection Current<br>• Power Consumption<br>• Analog-to-Digital Characteristics<br>• Digital-to-Analog Converter Characteristics<br>Added in new section:<br>• PTC Characteristics |


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**Table 64-9.** Revision E - 06/2019 (continued)

| Section Name or Type | Change Description |
|---|---|
| Electrical Characteristics at 105°C | Updated the following sections:<br>• Power Consumption<br>• Digital-to-Analog Converter Characteristics<br>• Analog-to-Digital Characteristics<br>Added new section:<br>• PTC Characteristics |
| Electrical Characteristics at 125°C | Updated the following sections:<br>• Power Consumption<br>• Analog-To-Digital Characteristics<br>• Digital-to-Analog Characteristics<br>Added new section:<br>• PTC Characteristics |

**Table 64-10.** Rev. D - 12/2018

| Section Name or Type | Change Description |
|---|---|
| Ordering Information | Added AEC-Q100 Qualified package type. |
| I/O Multiplexing and Considerations | Added information for GRXDV pin for 64-pin package devices. |
| AEC Q-100 Grade 1, 125°C Electrical Characteristics | Introduced device part numbers with AEC Q-100 Grade 1. |

**Table 64-11.** Rev. C - 11/2018

| Section Name or Type | Change Description |
|---|---|
| Ordering Information | Added ordering information for 105°C and 125°C temperature grade. |
| Pinout | Exposed pad info added for VQFN package. |
| I/O Multiplexing and Considerations | Corrected typographical errors for pin numbers PB19 and PB23. |
| Memories | Clarified NVM User Row size in table 9-1. |
| Processor and Architecture | Corrected typographical errors in section 10.2.2 Interrupt Line Mapping for SERCOMx interrupt line 7. |
| MCLK | Corrected typographical errors related to R/W bits for 15.8.8 APBA Mask Register. |
| PM | Updated Figure 18-2 Operating Conditions and SleepWalking to reflect that PL0 is not applicable to this product. |
| SUPC | Updated INTENCLR, INTENSET, INTFLAG, and STATUS Registers to reflect factory preprogramming of BOD12. |
| DMAC | Removed CHIP.ID information as it is not applicable to this product. |
| EVSYS | Corrected typographical errors in the USERm Register offset. |
| CCL | 1. Internal Events Inputs Selection (EVENT) section was updated by removing ASYNCEVENT related information.<br>2. Alternate 2 TC input source not applicable and was removed for LUTCTRL.INSELx bits. |
| ADC | Added clarification for INTREF to 45.8.6 Reference Control(REFCTRL). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-11.** Rev. C - 11/2018 (continued)

| Section Name or Type | Change Description |
|---|---|
| TC | 1. 48.7.1 Register Summary - 8-bit Mode<br>– Updated register bitfield with indexing to display usage.<br>2. 48.7.2 Register Summary - 16-bit Mode<br>a. Updated register bitfield with indexing to display usage.<br>b. Removed inapplicable register PER & PERBUF register information.<br>3. 48.7.3 Register Summary - 32-bit Mode<br>a. Updated register bitfield with indexing to display usage.<br>b. Removed inapplicable register PER & PERBUF register information. |
| TCC - Timer/ Counter for Control Applications | 1. Table 49-4. Output Matrix Channel Pin Routing Configuration updated to show all supported 6 capture channels.<br>2. Table 49-8. Fault and Capture Action updated by adding missing CAPTMARK value for CAPTURE bit fields.<br>3. Register INTENCLR, INTENSET, INTFLAG updated with missing UFS bit.<br>4. Removed unsupported bit info for the register 49.8.15 Pattern (PATT).<br>5. Missing POLx bits added to the register 49.8.16 Waveform (WAVE).<br>6. 49.7 Register Summary<br>– Updated register bitfield with indexing to display usage. |
| Electrical Characteristics at 85°C | 1. Clarified how $C_{LEXT}$ can be computed in section 54.12.1 Crystal Oscillator (XOSC) Characteristics and 54.12.2 External 32 kHz Crystal Oscillator (XOSC32K) Characteristics<br>2. Clarified capacitor requirements in Table 54-18. External Components Requirements in Switching Mode and Table 54-19 Decoupling Requirements.<br>3. Condition shown for VREF parameter is removed in table 54-24. Operating Conditions.<br>4. Conditions specified for table 54-29 Differential Mode is clarified for INL & DNL with Internal voltage reference.<br>5. Table 54-35. Flash Timing Characteristics is updated for Chip Erase maximum time.<br>6. Added the missing note in Table 54-44. Ultra-Low-Power Internal 32kHz Oscillator Electrical Characteristics.<br>7. Added the missing note in Table 54-48. Fractional Digital Phase Lock Loop Characteristics<br>8. Typo for the maximum value of tMOH in the Table 54-51. SPI Timing Characteristics and Requirements addressed.<br>9. Table 54-53. QSPI Maximum Frequency examples updated. |
| Electrical Characteristics at 105°C | Introduced device part numbers with Electrical Characteristics for 105°C temperature grade. |
| Electrical Characteristics at 125°C | Introduced device part numbers with Electrical Characteristics for 125°C temperature grade. |

**Table 64-12.** Rev. B - 4/2018

| Section Name or Type | Change Description |
|---|---|
| Features | Updated CAN FD reference.<br>Added 120-ball TFBGA package. |
| Configuration Summary | Added 120-ball TFBGA to the family feature tables. |
| Ordering Information | Updated the notes for devices in WLCSP packages.<br>Updated Package Type, adding CT = TFBGA. |
| Pinout | Added the 120-ball TFBGA package pinout diagram. |
| Multiplexed Signals | Added 120-ball TFBGA and updated Note 3 (see Table 6-1. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 64-12.** Rev. B - 4/2018 (continued)

| Section Name or Type | Change Description |
|---|---|
| OSC32KCTRL - 32 kHz Oscillators Controller | Added the EN1K and EN32K bits to the OSCULP32K register (see OSCULP32K). |
| SERCOM - Serial Communication Interface | Added Fractional Baud information to the Baud Rate Equations (see Table 33-2). |
| QSPI - Quad Serial Peripheral Interface | Added equations to the BAUD register (see BAUD). |
| CAN - Control Area Network | Updated the Overview.<br>Updated ISO 11898 references throughout the chapter. |
| Public Key Cryptography Controller (PUKCC) | Added the Public Key Cryptography Library (PUKCL) Application Programmer Interface (API) section. |
| TCC - Timer/Counter for Control Applications | Updated the number of TCC instances to 5 (4:0). |
| Electrical Characteristics at 85°C | (1) Improved SPI maximum speed information in Table 54-56.<br>(2). Added example for QSPI maximum frequency examples Table 54-58. |
| Packaging Information | Added the 120-ball TFBGA package (see 120-ball TFBGA). |

### Rev. A - 07/2017

This is the initial release of the document.


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.

# Microchip Information

## Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-3043-9

## Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

## Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## Product Page Links

ATSAMD51G18A, ATSAMD51G19A, ATSAMD51J18A, ATSAMD51J19A, ATSAMD51J20A, ATSAMD51N19A, ATSAMD51N20A, ATSAMD51P19A, ATSAMD51P20A, ATSAME51G18A, ATSAME51G19A, ATSAME51J18A, ATSAME51J19A, ATSAME51J20A, ATSAME51N19A, ATSAME51N20A, ATSAME53J18A, ATSAME53J19A, ATSAME53J20A, ATSAME53N19A, ATSAME53N20A, ATSAME54N19A, ATSAME54N20A, ATSAME54P19A, ATSAME54P20A


© 2026 Microchip Technology Inc. and its subsidiaries


Downloaded from Arrow.com.