# PIC32CM GC00 Family Data Sheet

## PIC32CM GC00

Product Page Links

# 32-bit Arm® Cortex®-M23 MCU with Functional Safety (FuSa), CAN-FD, Full-Speed USB, Enhanced Touch, and Advanced Analog

**Operating Conditions**

- 1.71V to 3.63V, -40°C to +125°C, DC to 48 MHz
- 1.71V to 3.63V, -40°C to +85°C, DC to 72 MHz

**Core: Arm Cortex-M23 CPU running at up to 72 MHz**

- Nested Vector Interrupt Controller (NVIC)
- Stack Limit Checking
- Memory Protection Unit (MPU)

**Memories**

- 128 KB SRAM Main Memory with ECC and fault injection capability
  - Retained in Idle, Standby and Hibernate modes
- 512 KB in-system self-programmable Flash (PFM) with Error Correction Code (ECC)
- Additional 16 KB of Boot Flash Memory (BFM)
- Additional 32 KB of Configuration Flash Memory (CFM)
- Flash Supports:
  - Error Correction Code (ECC) with fault injection capability
  - CRC of any contiguous section
  - Deep Power Down option while system is in Standby
  - In-band error reporting for both read and write accesses

**Safety and Compliance Standards**

- ISO 26262 ASIL-b
- IEC 60730
- AEC-Q100 Grade 1 (-40°C to +125°C)
- IEC 61508

**Advanced Analog Features and Touch**

- 12-bit ADC module:
  - Up to 4.5 Msps
  - Up to 12 single-ended channels or 3 differential + 6 single-ended channels
  - External reference support
- Two Analog Comparators with programmable voltage references
- Enhanced Peripheral Touch controller (PTC)
  - Up to 324 (18 x 18) mutual capacitance channels
  - Up to 36 self-capacitance channels with Driven Shield+ technology for better noise immunity and moisture tolerance
  - Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, and wheels
  - Hardware noise filtering and noise signal desynchronization for high conducted immunity
  - Supports wake-up on touch from Standby Sleep mode
  - Supports large self-capacitor sensor


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

**System**

- Integrated Power-on Reset (POR) and programmable Brown-out Detection (BOD) on VDDIO and VDDREG
- Programmable Low-Voltage Detect Module (LVD)
- 12-channel event system for Inter-peripheral Core-independent Operation (EVSYS)
- Unique 128-bit serial number

**Hardware Safety Features**

- ECC w/ Fault Injection capability on Flash and SRAM
- SRAM MBIST on boot User Mode accessible
- Peripheral registers write protection through PAC
- Fail-Safe Clock Monitor (CFD)
- ISO 26262:2018 Compliant

**Power Management**

- Idle mode for fast wake up time
- Standby mode, Backup mode, Off mode, and SleepWalking Peripherals
- Hibernate mode up to full SRAM retention

**Timers/Output Compare/Input Capture**

- Up to seven 16-bit Timer/Counters for Control (TCC), each with two double buffered compare/capture channels
- 32-bit Real-Time Counter (RTC) with clock/calendar functions
- Watchdog Timer (WDT) with Window mode

**Clock Management**

- 4 MHz to 48 MHz Crystal Oscillator (XOSC48)
  - Clock failure detection with safe clock switch
- 32.768 kHz crystal oscillator (XOSC32K)
  - Clock failure detection with safe clock switch
- 32.768 kHz ultra low-power internal RC oscillator (OSCULP32K)
- 48 MHz digital Phase-Locked Loop (DFLL48M)
- 1.6 GHz Phase Locked Loop (PLL)
- Frequency meter (FREQM)

**Software and Tools Support: Develop Prototypes Quickly with A Powerful, Easy-to-Use Ecosystem**

- Get code off to a head start with MPLAB Code Configurator
- Graphically configure peripherals, software libraries, and supported RTOS with MPLAB Harmony v3
- Download MPLAB XC Compiler
- Take advantage of MPLAB X IDE's support for 32-bit MCUs
- Select the best debugger for the project: MPLAB® ICE, MPLAB ICD, or PICkit™
- Zephyr, IAR, and Keil support

**Direct Memory Access (DMA)**

- Eight channels with 4 different block transfer modes
- Programmable 32-bit Cyclic Redundancy Check (CRC)

**Input/Output**

- High current pins with up to 25 mA source/sink
- Configurable Open-Drain Output on Digital I/O Pins
- Up to 10 5V-tolerant Input Pins (digital pins only)
- Up to 80 programmable I/O lines
- 16 external interrupts (EIC)
- Two Configurable Custom Logic (CCL) that supports:
  - Combinatorial logic functions, such as AND, NAND, OR, and NOR
  - Sequential logic functions, such as Flip-Flop and Latches
- One general purpose LDO output
  - 1.2/1.5/1.8/2.5V generated from VDDIO
  - Up to 100 mA

**Communication Interfaces /Digital Peripherals**

- Two CAN-FD modules (ISO 11898-1:2015)
  - Support CAN 2.0 A/B
- Six Serial Communication Interfaces (SERCOM), each configurable to operate as:
  - USART with full-duplex and single wire half-duplex configuration
  - $I^2C$ Host/Client up to 3.4 MHz
  - SPI
  - RS-485, IRDA, LIN Host/Client
- 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
  - Support crystal-less operation in device mode

**Debugger Development Support**

- In-circuit and in-application programming/debugging with SWD and JTAG
- Cortex-M debugger port
- Supports 8 breakpoints and 4 watch points
- IEEE®1149-compatible (JTAG) boundary scan
- Non-intrusive hardware-based instruction trace, Secure Debugging

**Table 1.** Packages

| Type | VQFN | | TQFP | | |
|---|---|---|---|---|---|
| Pin Count | 48 | 64 | 48 | 64 | 100 |
| I/O Pins (up to) | 36 | 50 | 36 | 50 | 80 |
| Contact/Lead Pitch | 0.5 mm | 0.5 mm | 0.5 mm | 0.5 mm | 0.5 mm |
| Dimensions | 7x7x0.9 mm | 9x9x1 mm | 7x7x1 mm | 10x10x1 mm | 14x14x1 mm |


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

# Table of Contents

32-bit Arm® Cortex®-M23 MCU with Functional Safety (FuSa), CAN-FD, Full-Speed USB, Enhanced Touch, and Advanced Analog ..... 1

1. Guidelines for Getting Started ..... 12
   - 1.1. Basic Connection Requirements ..... 12
   - 1.2. Decoupling Capacitors ..... 12
   - 1.3. External Reset (RESET) Pin ..... 13
   - 1.4. Debugging or Programming Pins ..... 13
   - 1.5. JTAG ..... 13
   - 1.6. External Oscillator Pins ..... 13
   - 1.7. Unused I/Os ..... 14
   - 1.8. Considerations When Interfacing to Remotely Powered Circuits ..... 15
   - 1.9. Designing for High-Speed Peripherals ..... 16
2. Configuration Summary ..... 19
3. Ordering Information ..... 20
4. Block Diagram ..... 21
5. Pinout and Packaging ..... 22
   - 5.1. 48-Pin TQFP and 48-Pin VQFN ..... 22
   - 5.2. 64-Pin TQFP and 64-Pin VQFN ..... 24
   - 5.3. 100-Pin TQFP ..... 27
6. Signal Description ..... 30
7. Power Supplies and Startup Considerations ..... 40
   - 7.1. Power Domain Overview ..... 40
   - 7.2. Power Domain Constraints ..... 40
   - 7.3. Device Startup ..... 42
   - 7.4. Analog Peripherals Considerations ..... 43
8. Product Mapping ..... 44
9. Peripherals ..... 47
   - 9.1. Register Description ..... 47
10. Processor and Architecture ..... 66
    - 10.1. Processor and Architecture ..... 66
    - 10.2. Arm Cortex-M23 Configuration ..... 66
    - 10.3. Nested Vector Interrupt Controller ..... 66
    - 10.4. NVIC Interrupt Mapping ..... 66
    - 10.5. High Speed Bus ..... 70
    - 10.6. Bus Matrix Connectivity ..... 70
11. Memories ..... 71
    - 11.1. Embedded Memories ..... 71
    - 11.2. Physical Memory Map ..... 71
    - 11.3. SRAM Memory Configuration ..... 71


© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

11.4. Flash Memories....71
11.5. Unique ID (UID)....72

12. Boot ROM....74
12.1. Boot ROM....74
12.2. Features....74
12.3. Block Diagram....74
12.4. Boot ROM Dependencies....75
12.5. Functional Description....76
12.6. Fuse Summary: ROMCFG....92
12.7. Fuse Summary: USERCFG....96

13. Multi-Channel RAM Controller (MCRAMC)....109
13.1. Overview....109
13.2. Features....109
13.3. Block Diagram....109
13.4. Functional Description....110
13.5. Peripheral Dependencies....112
13.6. Register Summary....113

14. Peripheral Access Controller (PAC)....124
14.1. Overview....124
14.2. Features....124
14.3. Block Diagram....124
14.4. Debug Operation....124
14.5. Register Access Protection....124
14.6. Functional Description....125
14.7. Peripheral Dependencies....126
14.8. Register Summary....127

15. Device Service Unit (DSU)....133
15.1. Overview....133
15.2. Features....133
15.3. Signal Description....133
15.4. DSU Block Diagram....134
15.5. Indexing....134
15.6. Debug Operation....134
15.7. Programming....138
15.8. Device Identification....139
15.9. Peripheral Dependencies....140
15.10. Register Summary....141

16. Clock Distribution System....176
16.1. Clock System....176

17. Oscillator Controller (OSCCTRL)....180
17.1. Overview....180
17.2. Features....180
17.3. OSCCTRL Block Diagram....181
17.4. Signal Descriptions....181


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

17.5. Peripheral Dependencies ... 182
17.6. Functional Description ... 183
17.7. Peripheral Dependencies ... 192
17.8. Register Summary ... 193

18. Generic Clock Controller (GCLK) ... 229
18.1. Overview ... 229
18.2. Features ... 229
18.3. Block Diagram ... 229
18.4. Signal Description ... 230
18.5. Functional Description ... 230
18.6. Peripheral Dependencies ... 235
18.7. GCLK Peripheral Channel Control Mapping ... 235
18.8. Register Summary ... 237

19. Main Clock (MCLK) ... 245
19.1. Overview ... 245
19.2. Features ... 245
19.3. Block Diagram ... 245
19.4. Functional Description ... 246
19.5. Peripheral Dependencies ... 247
19.6. MCLK AHB/APB Clock Enable Peripheral Mapping ... 248
19.7. Register Summary ... 250

20. 32 KHz Oscillators Controller (OSC32KCTRL) ... 256
20.1. Overview ... 256
20.2. Features ... 256
20.3. Block Diagram ... 257
20.4. Signal Description ... 257
20.5. Functional Description ... 257
20.6. Peripheral Dependencies ... 261
20.7. Register Summary ... 262

21. Watchdog Timer (WDT) ... 278
21.1. Overview ... 278
21.2. Features ... 278
21.3. Block Diagram ... 278
21.4. Functional Description ... 279
21.5. Peripheral Dependencies ... 283
21.6. Register Summary ... 284

22. Frequency Meter (FREQM) ... 295
22.1. Overview ... 295
22.2. Features ... 295
22.3. Block Diagram ... 295
22.4. Product Dependencies ... 295
22.5. Functional Description ... 296
22.6. Peripheral Dependencies ... 300
22.7. Register Summary ... 301


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

23. Real-Time Counter (RTC)..... 316
23.1. Overview..... 316
23.2. Features..... 316
23.3. Block Diagram..... 316
23.4. Signal Description..... 318
23.5. Peripheral Dependencies..... 318
23.6. Functional Description..... 319
23.7. Peripheral Dependencies..... 330
23.8. Register Summary - MODE0..... 331
23.9. Register Summary - MODE1..... 348
23.10. Register Summary - MODE2..... 367
23.11. Register Summary..... 384

24. Direct Memory Access Controller (DMAC)..... 394
24.1. Overview..... 394
24.2. Features..... 394
24.3. Block Diagram..... 396
24.4. Product Dependencies..... 396
24.5. Functional Description..... 397
24.6. Peripheral Dependencies..... 416
24.7. Register Summary..... 417
24.8. DMA Descriptor Register Summary..... 444

25. Supply Controller (SUPC)..... 451
25.1. Overview..... 451
25.2. Features..... 451
25.3. Block Diagram..... 452
25.4. Signals Description..... 453
25.5. Functional Description..... 453
25.6. Peripheral Dependencies..... 456
25.7. Register Summary..... 457

26. Power Manager (PM)..... 472
26.1. Overview..... 472
26.2. Features..... 472
26.3. Clocks..... 472
26.4. Interrupts..... 472
26.5. Debug Operation..... 472
26.6. Register Access Protection..... 473
26.7. Block Diagram..... 473
26.8. Functional Description..... 473
26.9. Peripheral Dependencies..... 481
26.10. Register Summary..... 482

27. Reset Controller (RSTC)..... 490
27.1. Overview..... 490
27.2. Features..... 490
27.3. Block Diagram..... 490
27.4. Signals Description..... 490
27.5. Functional Description..... 491


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

27.6. Peripheral Dependencies ... 492
27.7. Register Summary ... 493

28. External Interrupt Controller (EIC) ... 498
28.1. Overview ... 498
28.2. Features ... 498
28.3. Block Diagram ... 498
28.4. Signal Description ... 499
28.5. Product Dependencies ... 499
28.6. Functional Description ... 500
28.7. Sleep Mode Operation ... 504
28.8. Secure Access Right ... 504
28.9. Synchronization ... 505
28.10. Peripheral Dependencies ... 505
28.11. Register Summary ... 506

29. Non-Volatile Memory Controller (NVMCTRL) ... 521
29.1. Flash Controller, Write ... 521
29.2. Flash Controller, Read ... 534
29.3. Peripheral Dependencies ... 549
29.4. Peripheral Dependencies ... 550
29.5. Register Summary ... 551
29.6. Register Summary ... 575

30. Event System (EVSYS) ... 613
30.1. Overview ... 613
30.2. Features ... 613
30.3. Block Diagram ... 613
30.4. Power Management ... 614
30.5. Clocks ... 614
30.6. Functional Description ... 614
30.7. Peripheral Dependencies ... 620
30.8. Register Summary ... 621

31. I/O Pin Controller (PORT) ... 645
31.1. Overview ... 645
31.2. Features ... 645
31.3. Block Diagram ... 646
31.4. Signal Description ... 646
31.5. I/O Lines ... 646
31.6. Clocks ... 646
31.7. Power Management ... 646
31.8. Debug Operation ... 647
31.9. Register Access Protection ... 647
31.10. Analog Connections ... 647
31.11. Functional Description ... 647
31.12. Peripheral Dependencies ... 653
31.13. Register Summary ... 654

32. Serial Communication Interface (SERCOM) ... 674


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

32.1. Overview....674
32.2. Features....674
32.3. Block Diagram....674
32.4. Functional Description....675
32.5. Synchronous and Asynchronous Receiver and Transmitter (SERCOM USART)....679
32.6. SERCOM Serial Peripheral Interface (SPI)....696
32.7. SERCOM $I^2C$....713
32.8. Peripheral Dependencies....732
32.9. Register Summary - USART....734
32.10. Register Summary - SPI....764
32.11. Register Summary - I2CM....789
32.12. Register Summary - I2CS....817

33. Controller Area Network (CAN)....843
33.1. Overview....843
33.2. Features....843
33.3. Block Diagram....844
33.4. Signal Description....844
33.5. Functional Description....844
33.6. Message RAM....865
33.7. Peripheral Dependencies....867
33.8. Register Summary....868
33.9. Rx Buffer and FIFO Element Register Summary....974
33.10. Tx Buffer Element Register Summary....979
33.11. Tx Event FIFO Element Register Summary....984
33.12. Standard Message ID Filter Element Register Summary....987
33.13. Extended Message ID Filter Element Register Summary....989

34. Universal Serial Bus (USB)....992
34.1. Overview....992
34.2. Features....992
34.3. USB Block Diagram....993
34.4. Signal Description....993
34.5. Product Dependancies....993
34.6. Functional Description....994
34.7. Peripheral Dependencies....1012
34.8. Common Device Host Register Summary....1013
34.9. Register Summary - DEVICE....1020
34.10. Device Endpoint Register Summary....1033
34.11. USB Descriptor Register Summary....1042
34.12. Register Summary - HOST....1048
34.13. Host Registers - Pipe - Register Summary....1062
34.14. Host Registers - Pipe RAM Summary....1074

35. Configurable Custom Logic (CCL)....1083
35.1. Overview....1083
35.2. Features....1083
35.3. Block Diagram....1084
35.4. Signal Description....1084


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

35.5. I/O Lines....1084
35.6. Power Management....1084
35.7. Clocks....1084
35.8. Events....1085
35.9. Debug Operation....1085
35.10. Register Access Protection....1085
35.11. Functional Description....1085
35.12. Events....1093
35.13. Sleep Mode Operation....1093
35.14. Peripheral Dependencies....1093
35.15. Register Summary....1095

36. Analog to Digital Converter (ADC)....1101
36.1. Overview....1101
36.2. Features....1101
36.3. Block Diagram....1102
36.4. Signal Description....1105
36.5. Functional Description....1105
36.6. Peripheral Dependencies....1114
36.7. Register Summary....1115

37. Analog Comparator (AC)....1171
37.1. Overview....1171
37.2. Features....1171
37.3. Block Diagram....1172
37.4. Analog Connections....1172
37.5. Functional Description....1173
37.6. Peripheral Dependencies....1181
37.7. Register Summary....1182

38. Timer/Counter for Control Applications (TCC)....1205
38.1. Overview....1205
38.2. Features....1205
38.3. Block Diagram....1206
38.4. Signal Description....1207
38.5. Functional Description....1207
38.6. Peripheral Dependencies....1246
38.7. Register Summary....1250

39. Peripheral Touch Controller (PTC)....1289
39.1. Overview....1289
39.2. Features....1289
39.3. Block Diagram....1290
39.4. Signal Description....1290
39.5. Peripheral Dependencies....1291
39.6. Functional Description....1292
39.7. Peripheral Dependencies....1303

40. Electrical Characteristics....1304
40.1. Absolute Maximum Electrical Characteristics....1304


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

40.2. Operating Frequencies and Thermal Specifications ..... 1305
40.3. Power Supply ..... 1305
40.4. Active Power ..... 1308
40.5. Idle Power ..... 1310
40.6. Standby Power ..... 1311
40.7. Hibernate Power ..... 1312
40.8. Backup Power ..... 1314
40.9. OFF Power ..... 1315
40.10. Wake-Up Timing ..... 1316
40.11. Peripheral Active Power ..... 1316
40.12. I/O Pin Electrical Characteristics ..... 1317
40.13. Internal Voltage Reference Electrical Specifications (IVREF) ..... 1323
40.14. Maximum Clock Frequencies ..... 1324
40.15. XOSC Electrical Characteristics ..... 1325
40.16. XOSC32K Electrical Characteristics ..... 1328
40.17. OSCULP32K Electrical Characteristics ..... 1330
40.18. DFLL-FDPLL Electrical Characteristics ..... 1330
40.19. Analog-to-Digital Converter (ADC) Electrical Specifications ..... 1333
40.20. Analog Comparator (AC) Electrical Characteristics ..... 1336
40.21. Peripheral Touch Controller (PTC) Electrical Characteristics ..... 1338
40.22. Serial Peripheral Interface (SPI) Timings Electrical Characteristics ..... 1339
40.23. Universal Asynchronous Receiver/Transmitter (UART) Electrical Characteristics ..... 1345
40.24. $I^2C$ Electrical Characteristics ..... 1346
40.25. Controller Area Network (CAN) Electrical Characteristics ..... 1353
40.26. Timer/Counter Controller (TCC) Electrical Characteristics ..... 1353
40.27. Universal Serial Bus (USB) Electrical Characteristics ..... 1355
40.28. Flash Controller Electrical Characteristics ..... 1355
40.29. Frequency Meter (FREQM) Electrical Characteristics ..... 1356
40.30. JTAG Electrical Characteristics ..... 1357
40.31. SWD 2-Wire Electrical Characteristics ..... 1358
40.32. Configurable Custom Logic (CCL) Electrical Specifications ..... 1358
40.33. User LDO Electrical Specifications ..... 1359
40.34. Multi-Channel Ram Controller (MCRAMC) Electrical Specifications ..... 1359

41. Electrical Characteristics 125°C ..... 1360
41.1. Absolute Maximum Ratings ..... 1360
41.2. Operating Frequencies and Thermal Specifications ..... 1361
41.3. MCU Active Power ..... 1361
41.4. Idle Power ..... 1363
41.5. Standby Power ..... 1365
41.6. Hibernate Power ..... 1367
41.7. Backup Power ..... 1368
41.8. OFF Power ..... 1369
41.9. Maximum Clock Frequencies ..... 1370
41.10. Analog Comparator (AC) Electrical Specifications ..... 1371
41.11. Serial Peripheral Interface (SPI) Timings and Electrical Characteristics ..... 1372
41.12. Timer/Counter Controller (TCC) Electrical Characteristics ..... 1378
41.13. JTAG Electrical Specifications ..... 1380


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

42. Packaging Information..... 1381
42.1. Package Marking Information..... 1381
42.2. Package Drawings..... 1381
42.3. Soldering Profile..... 1399
43. Schematic Checklist..... 1401
43.1. Introduction..... 1401
44. Common Conventions..... 1402
44.1. Numerical Notation..... 1402
44.2. Memory Size and Type..... 1402
44.3. Frequency and Time..... 1402
44.4. Registers and Bits..... 1402
45. Acronyms and Abbreviations..... 1404
46. Revision History..... 1406
Microchip Information..... 1407
Trademarks..... 1407
Legal Notice..... 1407
Microchip Devices Code Protection Feature..... 1407
Product Page Links..... 1408


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

# 1. Guidelines for Getting Started

## 1.1. Basic Connection Requirements

Getting started with the PIC32CM GC00 family of 32-bit Microcontrollers (MCUs) requires attention to a minimal set of device pin connections before proceeding with development. The following is a list of pin names, which must be connected always:

- All VDDIO and VSS pins (see Decoupling Capacitors)
- All AVDD and AVSS pins, even if the ADC module is not used (see Decoupling Capacitors)
- The $\overline{\text{RESET}}$ pin (see External $\overline{\text{RESET}}$ Pin)
- The SWDIO and SWCLK pins, used for In-Circuit Programming and debugging purposes (see Debug/Programming Pins)
- The XIN/XOUT and XIN32/XOUT32 pins, when external oscillator source is used (see External Oscillator Pins)

The following pins may be required:

- The ADC_VREFH pin is used when an external voltage reference for the ADC module is implemented
- TDI, TMS, TDO, and TCK if JTAG is used

## 1.2. Decoupling Capacitors

The use of decoupling capacitors on power supply pins, such as VREG, VDDIO, and AVDD is required, refer to the Schematic Checklist chapter.

Consider the following criteria when using decoupling capacitors:

- **Bulk capacitors:** Must be utilized on all power pins as indicated in the Schematic Checklist chapter.
- **Value and type of capacitor:** A value of 0.1 µF (100 nF), 10-20V is recommended. The capacitor should be a low Equivalent Series Resistance (low-ESR) capacitor and have resonance frequency in the range of 20 MHz and higher. It is further recommended that ceramic capacitors be used.
- **Placement on the printed circuit board:** The decoupling capacitors should be placed as close to the pins as possible. It is recommended that the capacitors be placed on the same side of the board as the device. If space is constricted, the capacitor can be placed on another layer on the PCB using a via; however, ensure that the trace length from the pin to the capacitor is within one-quarter inch (6 mm) in length.
- **Handling high frequency noise:** If the board is experiencing high frequency noise, upward of tens of MHz, add a second ceramic-type capacitor in parallel to the above described decoupling capacitor. The value of the second capacitor can be in the range of 0.01 µF to 0.001 µF. Place this second capacitor next to the primary decoupling capacitor. In high-speed circuit designs, consider implementing a decade pair of capacitances as close to the power and ground pins as possible. For example, 0.1 µF in parallel with 0.001 µF.
- **Maximizing performance:** On the board layout from the power supply circuit, run the power and return traces to the decoupling capacitors first, and then to the device pins. This ensures that the decoupling capacitors are first in the power chain. Equally important is to keep the trace length between the capacitor and the power pins to a minimum thereby reducing PCB track inductance.

### 1.2.1. Bulk Capacitors

The use of a bulk capacitor is recommended to improve power supply stability. Typical values range from 4.7 µF to 22 µF ceramic or tantalum capacitors with low ESR. This capacitor should be located


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

as close to the device as possible. For additional information, refer to the Schematic Checklist chapter.

## 1.3. External Reset ($\overline{\text{RESET}}$) Pin

Pulling the $\overline{\text{RESET}}$ pin low generates a device Reset. For example, as illustrated in the Schematic Checklist chapter, it is recommended that the capacitor C, be isolated from the $\overline{\text{RESET}}$ pin. Place the components illustrated within one-half inch (12 mm) from the $\overline{\text{RESET}}$ pin.

## 1.4. Debugging or Programming Pins

The SWDIO and SWCLK pins are used for In-Circuit programming and debugging purposes. It is recommended to keep the trace length between the debug external connector and the debug pins on the device as short as possible to minimize ESD/EMI vulnerabilities. If the debug external connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ωs, not to exceed 100 Ωs with protection using Transient Voltage Suppressors (TVS), at the user's discretion. For additional information, refer to the Schematic Checklist chapter.

## 1.5. JTAG

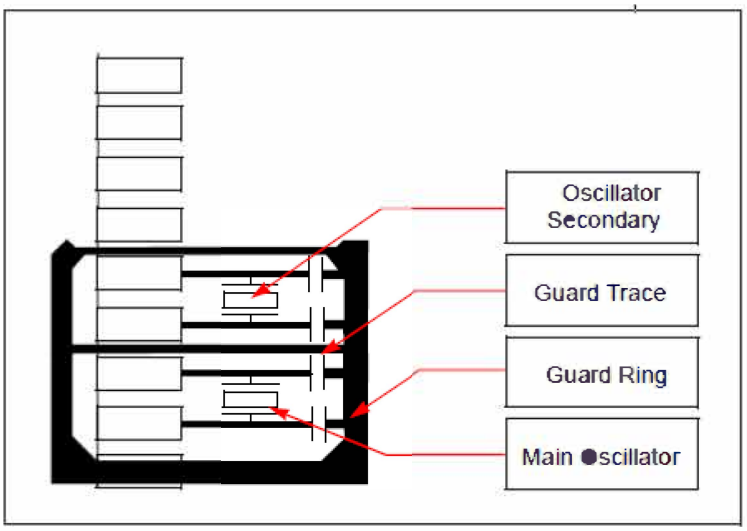
The TMS, TDO, TDI and TCK pins are used for testing, debugging, and boundary scan according to the Joint Test Action Group (JTAG) standard. It is recommended to keep the trace length between the JTAG connector and the JTAG pins on the device as short as possible. If the JTAG connector is expected to experience an ESD event, a series resistor is recommended, with the value in the range of a few tens of Ωs, not to exceed 100 Ωs with protection using Transient Voltage Suppressors (TVS), at the user's discretion. For additional information, refer to the Schematic Checklist chapter.

## 1.6. External Oscillator Pins

Many MCUs have options for at least two oscillators: a high-frequency primary oscillator and a low-frequency secondary oscillator.

The oscillator circuit should be placed on the same side of the board as the device. Also, place the oscillator circuit close to the respective oscillator pins, not exceeding one-half inch (12 mm) distance between them. The load capacitors should be placed next to the oscillator itself, on the same side of the board. Use a grounded copper pour around the oscillator circuit to isolate them from surrounding circuits. The grounded copper pour should be routed directly to the MCU ground. Do not run any signal traces or power traces inside the ground pour. Also, if using a two-sided board, avoid any traces on the other side of the board where the crystal is placed. A suggested layout is illustrated in the following figure.

**Figure 1-1.** Suggested Oscillator Circuit Placement





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

### 1.6.1. Crystal Oscillator Design Consideration

The following hypothetical example assumptions are used to calculate the Primary Oscillator loading capacitor values:

- CIN = XIN Pin Capacitance = 4 pF
- COUT = XOUT Pin Capacitance = 4 pF
- PCB stray capacitance (i.e., 12 mm length) = 2.5 pF
- C1 and C2 = The loading capacitors to use on your crystal circuit design to guarantee that the effective capacitance as seen by the crystal in circuit meets the crystal manufacturer specification MFG Crystal Data Sheet CLOAD specification:

CLOAD = {( [CIN + C1] * [COUT + C2] ) / [CIN + C1 + C2 + COUT] } + oscillator PCB stray capacitance

**Hypothetical Example Crystal Load Capacitor Calculation**

Crystal manufacturer data sheet spec example: *CLOAD* = 15 *pF*

Therefore,

*MFG CLOAD* = {( [*CIN* + *C1*] * [*COUT* + *C2*] ) / [*CIN* + *C1* + *C2* + *COUT*] } + *estimated oscillator PCB stray capacitance.*

Assuming *C1* = *C2* and PIC32C *Cin* = *Cout*, the formula can be further simplified and restated to solve for *C1* and *C2* by: *C1 = C2 = ((2 * MFG CLoad spec) - Cin - (2 * PCB capacitance)) = ((2 * 15) - 4 - (2 * 2.5 pF)) = (30 - 4 - 5) = 21 pF*

Therefore,

*C1 = C2 = 21 pF* is the correct loading capacitors to use on your crystal circuit design to guarantee that the effective capacitance as seen by the crystal in circuit in this example is 15 pF to meet the crystal manufacturer specification.

**Note:** Do not add excessive gain such that the oscillator signal is clipped, flat on top of the sine wave. If so, you need to reduce the gain accordingly. Failure to do so can stress and age the crystal, which can result in an early failure. When measuring the oscillator signal you must use an FET active-powered scope probe with 1 pF or the scope probe itself will unduly change the gain and peak-to-peak levels.

## 1.7. Unused I/Os

Unused I/O pins should not be allowed to float as inputs. It is recommended that unused inputs be ganged together and connecting through a 1k resistor to digital ground or individually or in multiple groups through 1k as the PCB layout permits. This minimizes the chip vulnerability to ESD and radiated EMI due to a high voltage discharge event.

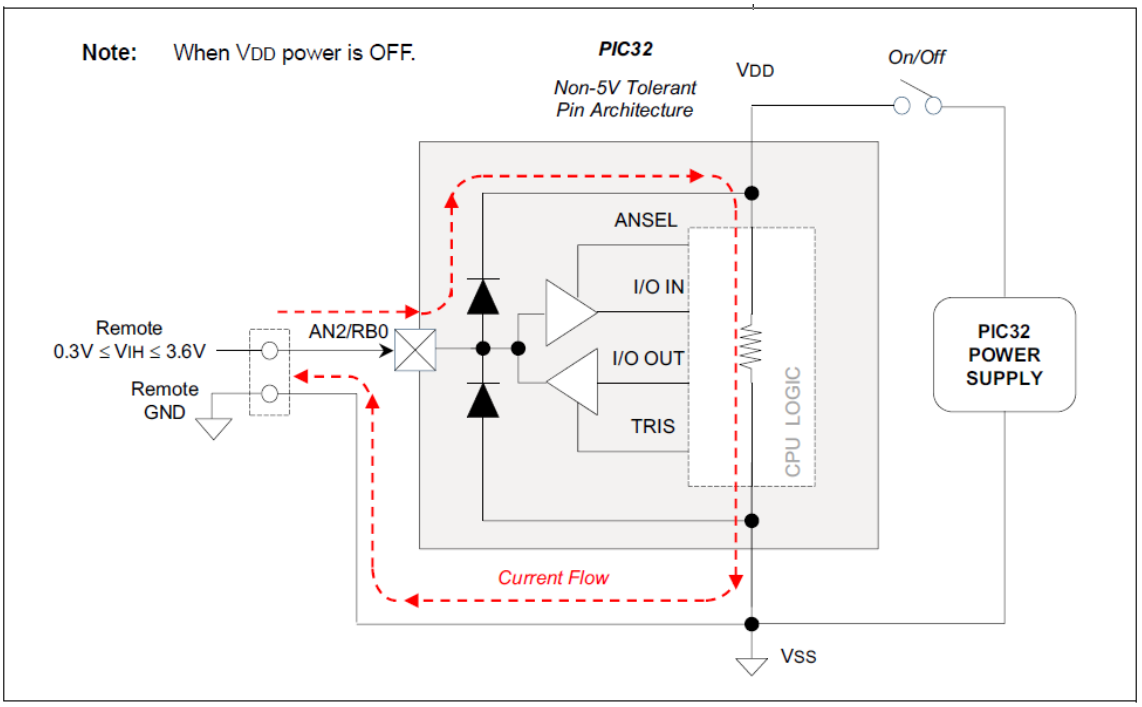

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 1.8. Considerations When Interfacing to Remotely Powered Circuits

### 1.8.1. Non-5V Tolerant Input Pins

A quick review of the absolute maximum rating in the *Electrical Characteristics (85°C)* will indicate that the voltage on any non-5V tolerant pin may not exceed VDD + 0.4V unless the input current is limited to meet the respective injection current specifications defined by electrical spec parameters DI_19, DI_21, and DI_23 in *I/O Pin Electrical Specifications*. The following figure shows an example of a remote circuit using an independent power source, which is powered while connected to a PIC32C non-5V tolerant circuit that is not powered.

**Figure 1-2.** PIC32C Non-5V Tolerant I/O Pin Circuit Example




Without proper signal isolation, on non-5V tolerant pins, the remote signal can power the PIC32C device through the high side ESD protection diodes. Besides violating the absolute maximum rating specification when VDD of the PIC32C device is restored and ramping up or ramping down, it can also negatively affect the internal Power-on Reset (POR) and Brown-out Reset (BOR) circuits, which can lead to improper initialization of internal PIC32C logic circuits. In these cases, it is recommended to implement digital or analog signal isolation as depicted in the following figure, as appropriate or use series I/O pin resistors when possible to limit the injection current to less than 5 mA/pin and/or 20 mA total for the whole MCU. This is indicative of all industry microcontrollers and not just Microchip products.


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

**Figure 1-3.** Example Digital/Analog Signal Isolation Circuits

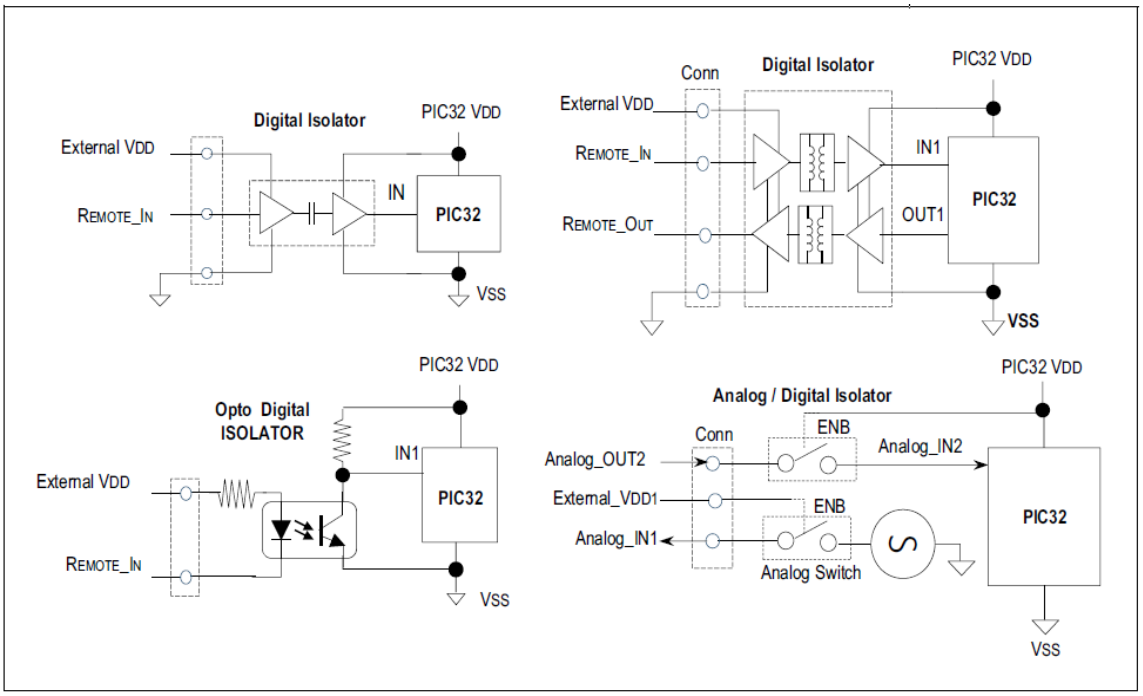



**Table 1-1.** Examples of Digital/Analog Isolators with Optional Level Translation

| Example Digital/ Analog Signal Isolation Circuits | Inductive Coupling | Capacitive Coupling | Opto Coupling | Analog/Digital Switch |
|---|---|---|---|---|
| ADuM7241/40 ARZ (1 Mbps) | X | — | — | — |
| ADuM7241/40 CRZ (25 Mbps) | X | — | — | — |
| ISO721 | — | X | — | — |
| LTV-829S (2 Channel) | — | — | X | — |
| LTV-849S (4 Channel) | — | — | X | — |
| FSA266/NC7WB66 | — | — | — | X |

## 1.9. Designing for High-Speed Peripherals

The PIC32C family of devices have peripherals that operate at frequencies much higher than typical for an embedded environment.

Due to these high-speed peripheral signals, it is important to consider several factors when designing a product that uses these peripherals, as well as the PCB on which these components will be placed. Adhering to these recommendations will help achieve the following goals:

- Minimize the effects of electromagnetic interference for the proper operation of the product.
- Run all PCB high-speed signals first on component side of PCB.
- Ensure signals arrive at their intended destination simultaneously by matching critical trace lengths on the PCB.


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

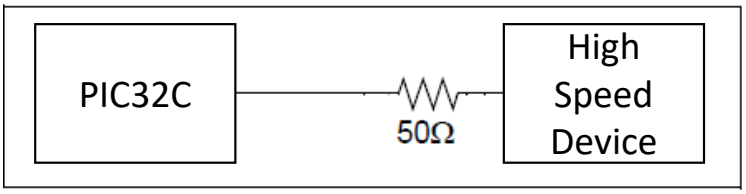
- Minimize crosstalk. Insure continuous ground under all high-speed signals.
- Maintain signal integrity by the use of termination resistors in the 30-50 Ω range.
- Reduce system noise by using bulk and high frequency decoupling caps and inductors on power rails.
- Minimize ground bounce and power sag. Use a dedicated ground plane if possible or at a minimum a star ground configuration. Do not daisy chain ground and power traces to components.

### 1.9.1. System Design

#### 1.9.1.1. Impedance Matching

When selecting parts to place on the high-speed signal bus, if the remote I/O pin impedance of the peripheral device does not match the impedance of the pins on the PIC32C device to which it is connected, signal reflections could result, thereby degrading the quality of the signal. If it is not possible to select a product that matches impedance, place a series resistor at the load to create the matching impedance, as shown in the following figure.

**Figure 1-4.** Series Resistor





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

#### 1.9.1.2. PCB Layout Recommendations

The following recommendations will help ensure the PCB layout will promote the goals previously listed.

- **Component Placement:**
  - Place bypass capacitors as close to their component power and ground pins as possible, and place them on the same side of the PCB.
  - Devices on the same bus that have larger setup times must be placed closer to the PIC32CM GC00 family of devices.
- **Power and Ground:**
  - Multi-layer PCBs will allow separate power and ground planes
  - Each ground pin should be connected to the ground plane individually
  - Place bypass capacitor vias as close to the pad as possible (preferably inside the pad)
  - If power and ground planes are not used, maximize the width for power and ground traces
  - Use low-ESR, surface-mount bypass capacitors
- **Clocks and Oscillators:**
  - Place crystals as close as possible to the XIN or XOUT pin of the PIC32CM GC00 family of devices
  - Do not route high-speed signals near the clock or oscillator
  - Avoid via usage and branches in high-speed clock lines
  - Place termination resistors at the end of clock lines
- **Traces:**
  - Higher-priority signals must have the shortest traces
  - Avoid long run lengths on parallel traces to reduce coupling
  - Make the clock traces as straight as possible
  - Use rounded turns rather than right-angle turns
  - Have traces on different layers intersect on right angles to minimize crosstalk
  - Maximize the distance between traces, preferably no less than three times the trace width
  - Power traces should be as short and as wide as possible
  - High-speed traces must have a continuous ground beneath them

#### 1.9.1.3. EMI/EMC/EFT (IEC 61000-4-4 and IEC 61000-4-2) Supression Considerations

The use of LDO regulators is preferred to reduce overall system noise and provide a cleaner power source. However, when utilizing switching buck/boost regulators as the local power source for PIC32CM GC00 devices, as well as in electrically noisy environments or test conditions required for IEC 61000-4-4 and IEC 61000-4-2, users should evaluate the use of Pie-Filters (i.e., L-C) on the power pins, as described in the *Power Supplies* chapter. In addition to a less noisy power source, use of this type of T-Filter can greatly reduce susceptibility to EMI sources and events. Use Transient Voltage Suppressors (TVS) on power buses as well as on all external PCB signal connections. If design requirements mandate the use of a buck/boost regulators be sure inductor used is shielded type.


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

# 2. Configuration Summary

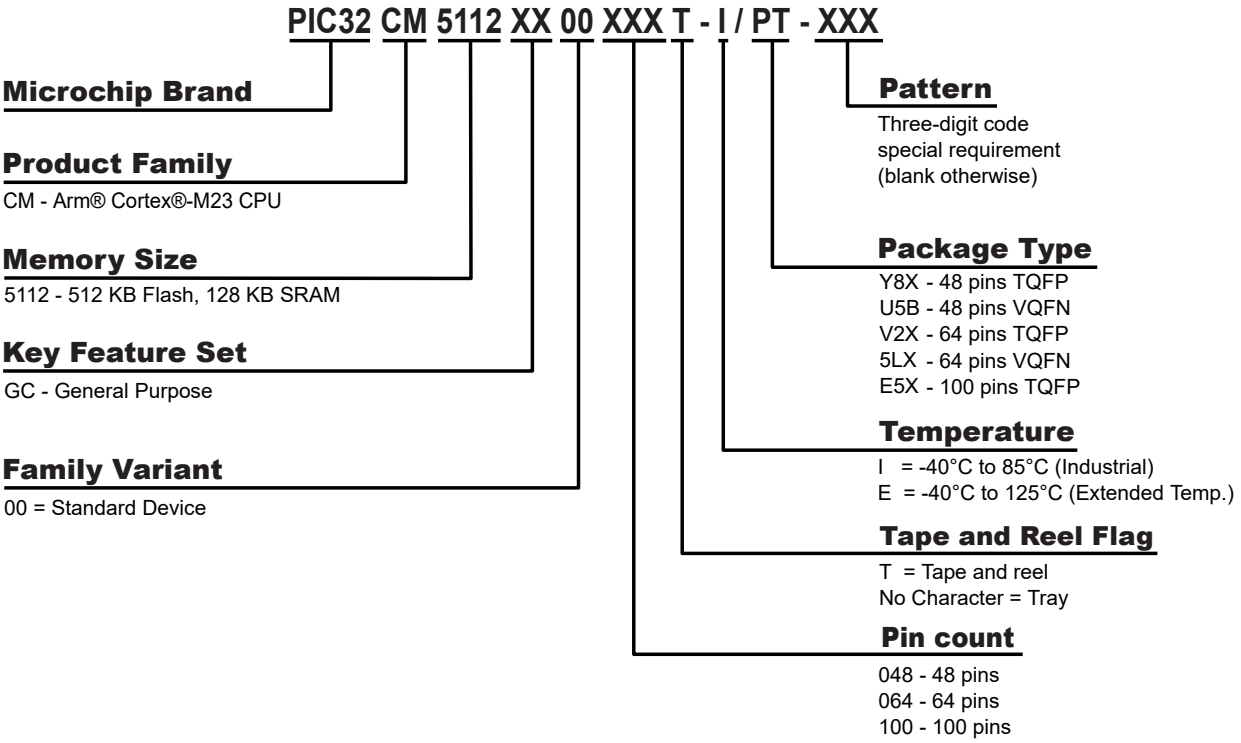
**Table 2-1.** PIC32CM GC00 Family Features

| Part Number | Pin Count | FLASH | SRAM | ADC | ADC Channels | Analog Comparators | TCC | SERCOM | ISO CAN_FD | USB FS | RTC | PTC | CCL | DMA Channels | WDT | FREQM | JTAG/ SWD /Trace | I/O Pins |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC32CM5112GC00048 | 48 | 512 | 128 | 1 | 12 | 2 | 7 | 6 | 2 | Y | Y | Y | 2 | 8 | Y | Y | Y | 36 |
| PIC32CM5112GC00064 | 64 | 512 | 128 | 1 | 12 | 2 | 7 | 6 | 2 | Y | Y | Y | 2 | 8 | Y | Y | Y | 50 |
| PIC32CM5112GC00100 | 100 | 512 | 128 | 1 | 12 | 2 | 7 | 6 | 2 | Y | Y | Y | 2 | 8 | Y | Y | Y | 80 |

© 2026 Microchip Technology Inc. and its subsidiaries

Downloaded from Arrow.com.

# 3. Ordering Information



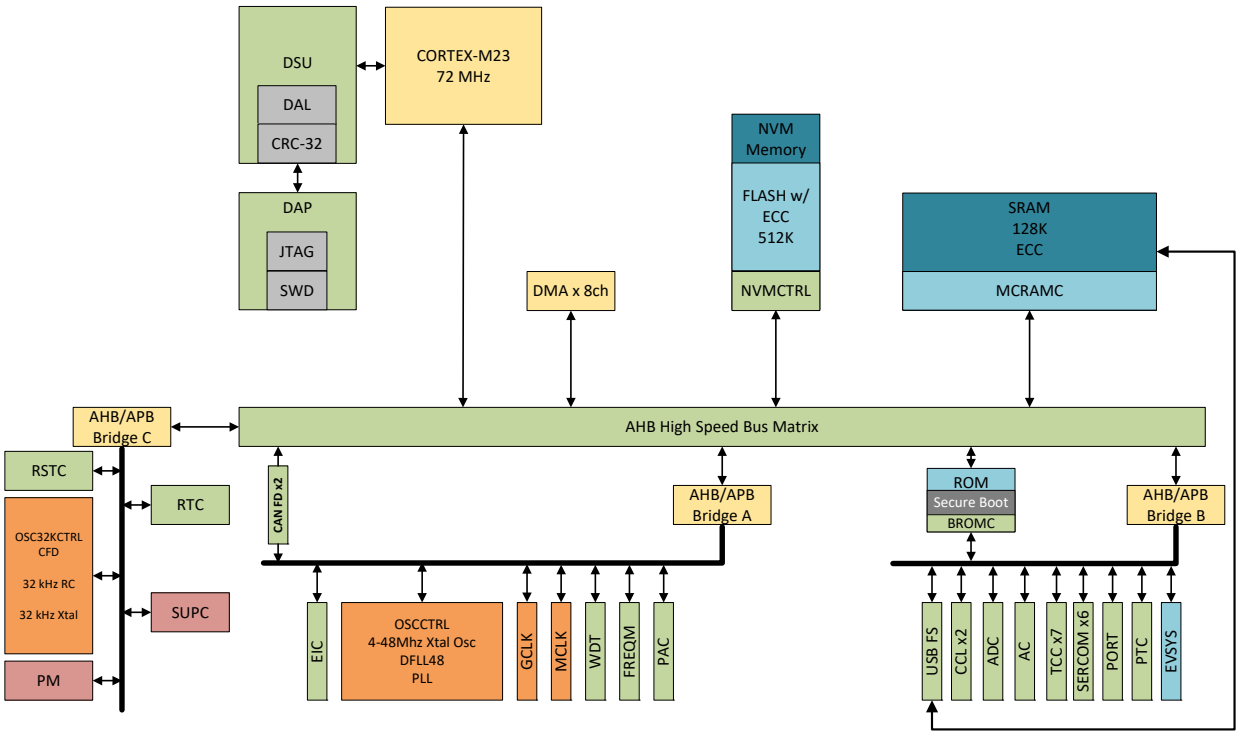

**Notes:**

1. Extended Temperature devices are AEC-Q100 Grade 1 Qualified.
2. The VQFN packages are with wettable flanks.


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

# 4. Block Diagram

**Figure 4-1.** PIC32CM GC00 Block Diagram





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

# 5. Pinout and Packaging

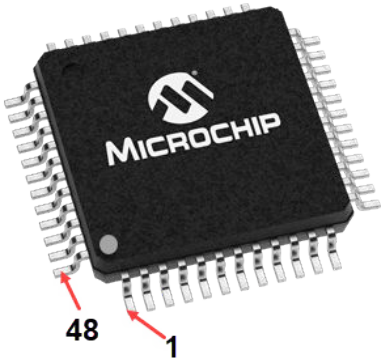
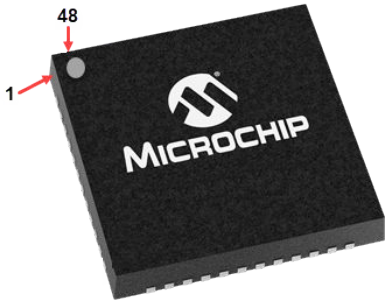
Each pin is controlled by the I/O Pin Controller (PORT) as a general purpose I/O and alternatively can be assigned to one of the peripheral functions listed in the PMUX Values columns.

The following tables describe the peripheral signals multiplexed to the PORT I/O pins for each package.

The column “Reset State” indicates the reset state of the line with mnemonics:

- "I/O" or "Priority Function" indicates whether the I/O pin resets in I/O mode or in peripheral function mode.
- “I”/”O”/"Hi-Z" indicates whether the I/O is configured as an input, output, or is tri-stated.
- “PU”/“PD” indicates whether pull up, pull down, or nothing is enabled.

## 5.1. 48-Pin TQFP and 48-Pin VQFN

| 48-Pin VQFN and 48-Pin TQFP (Top View) (1) | |
|---|---|
| PIC32CM5112GC00048 | TQFP<br>  <br>VQFN<br>  |

**Note:**

1. The VQFN package has wettable flanks.

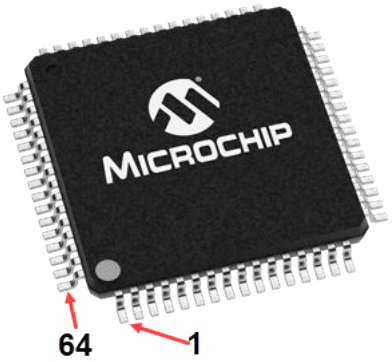
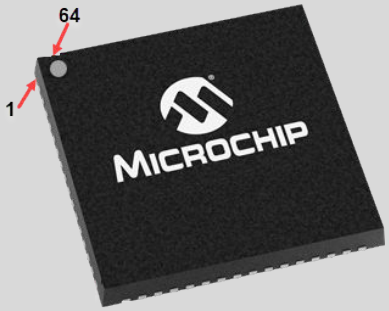

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 5-1.** 48-Pin VQFN and 48-Pin TQFP

| Pin | MUXEN = 0 | Priority Function | MUXEN = 1 | | | | | | | | | | Supply | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PMUX Values | | | | | | | | | | | |
| | | | 0x0 | 0x1 | | | 0x3 | 0x5 | 0x7 | 0x8 | 0xA | 0xF | | |
| | | | EIC | AC | ADC | ADCN | SERCOM | TCC | CAN | CCL | GCLK | PTC | | |
| 1 | PA00 | - | EXTINT[0] | - | - | - | SERCOM3/PAD[1] | TCC5/WO[1] | CAN1/RX | CCL0/IN[6] | - | XY[4] | VDDIO | Hi-Z |
| 2 | PA01 | - | EXTINT[1] | CMP[0] | - | - | SERCOM3/PAD[2] | TCC6/WO[0] | CAN1/TX | CCL0/IN[7] | - | XY[5] | VDDIO | Hi-Z |
| 3 | PA02 | - | EXTINT[2] | CMP[1] | - | - | SERCOM3/PAD[3] | TCC6/WO[1] | - | CCL0/OUT[2] | - | XY[6] | VDDIO | Hi-Z |
| 4 | PA03 | - | EXTINT[3] | - | AIN[0] | - | SERCOM4/PAD[0] | - | - | - | - | XY[7] | VDDIO | Hi-Z |
| 5 | PA04 | - | EXTINT[4] | - | AIN[1] | ANN[0] | SERCOM4/PAD[1] | - | - | - | - | XY[8] | VDDIO | Hi-Z |
| 6 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 7 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 8 | PA05 | - | EXTINT[5] | - | AIN[2] | - | SERCOM4/PAD[2] | - | - | CCL1/OUT[2] | - | XY[9] | VDDIO | Hi-Z |
| 9 | PA06 | - | EXTINT[6] | - | AIN[3] | ANN[2] | SERCOM4/PAD[3] | - | - | CCL1/IN[7] | - | XY[10] | VDDIO | Hi-Z |
| 10 | PA07 | RTC/IN[0] | EXTINT[7] | AIN[0] | AIN[4] | - | - | - | - | - | - | XY[11] | VDDIO | Hi-Z |
| 11 | PA08 | RTC/IN[1] | EXTINT[8] | AIN[1] | AIN[5] | ANN[4] | - | - | - | - | - | XY[12] | VDDIO | Hi-Z |
| 12 | PA09 | RTC/IN[2] | EXTINT[9] | - | AIN[6] | VREFH | - | - | - | CCL1/IN[6] | - | XY[13] | VDDIO | Hi-Z |
| 13 | PB00 | RTC/IN[3] | EXTINT[0] | AIN[2] | AIN[7] | - | - | - | - | CCL1/OUT[3] | - | XY[14] | VDDIO | Hi-Z |
| 14 | AVDD | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 15 | AVSS/VSSREG | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 16 | PB01 | XIN32 | EXTINT[1] | - | - | - | - | - | - | - | - | - | VDDIO | Hi-Z |
| 17 | PB02 | XOUT32 | EXTINT[2] | - | - | - | - | - | - | - | - | - | VDDIO | Hi-Z |
| 18 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 19 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 20 | VDDREG | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 21 | PB03 | RTC/OUT[0] | EXTINT[3] | AIN[3] | AIN[8] | - | SERCOM5/PAD[0] | - | - | CCL1/IN[10] | - | XY[15] | VDDIO | Hi-Z |
| 22 | PB04 | RTC/OUT[1], SUPC/OUT[0] | EXTINT[4] | - | AIN[9] | - | SERCOM5/PAD[1] | - | - | CCL1/IN[9] | - | XY[16] | VDDIO | Hi-Z |
| 23 | PB05 | RTC/OUT[2], SUPC/OUT[1] | EXTINT[5] | - | AIN[10] | - | SERCOM5/PAD[2] | - | - | CCL0/IN[9] | - | XY[17] | VDDIO | Hi-Z |
| 24 | PB06 | RTC/OUT[3] | EXTINT[6] | - | AIN[11] | - | SERCOM5/PAD[3] | - | - | CCL0/IN[10] | IO[0] | XY[18] | VDDIO | Hi-Z |
| 25 | PC00 | - | EXTINT[0] | - | - | - | SERCOM0/PAD[0] | TCC0/WO[0] | - | CCL0/OUT[3] | IO[1] | XY[19] | VDDIO | Hi-Z |
| 26 | PC01 | - | EXTINT[1] | - | - | - | SERCOM0/PAD[1] | TCC0/WO[1] | CAN0/RX | CCL0/IN[0] | IO[2] | XY[20] | VDDIO | Hi-Z |
| 27 | PC02 | - | EXTINT[2] | - | - | - | SERCOM0/PAD[2] | TCC1/WO[0] | CAN0/TX | CCL0/IN[1] | IO[3] | XY[21] | VDDIO | Hi-Z |
| 28 | PC03 | - | EXTINT[3] | - | - | - | SERCOM0/PAD[3] | TCC1/WO[1] | - | CCL0/OUT[0] | IO[4] | ECI[0] | VDDIO | Hi-Z |
| 29 | PC04 | - | EXTINT[4] | - | - | - | SERCOM2/PAD[0] | TCC2/WO[0] | - | CCL1/IN[0] | IO[5] | ECI[1] | VDDIO | Hi-Z |
| 30 | $\overline{\text{RESET}}$ | - | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 31 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 32 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 33 | PC05 | TDO | EXTINT[5] | - | - | - | SERCOM1/PAD[3] | TCC2/WO[1] | CAN1/RX | CCL1/IN[1] | IO[1] | - | VDDIO | Hi-Z |
| 34 | PC06 | TDI | EXTINT[6] | - | - | - | SERCOM1/PAD[2] | - | CAN1/TX | CCL1/OUT[0] | - | - | VDDIO | Hi-Z |
| 35 | - | TMS/SWDIO | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 36 | - | TCK/SWCLK | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 37 | PD00 | - | NMI | - | - | - | SERCOM1/PAD[1] | TCC3/WO[0] | - | CCL0/IN[3] | - | - | VDDIO | Hi-Z |
| 38 | PD01 | - | EXTINT[1] | - | - | - | SERCOM1/PAD[0] | TCC3/WO[1] | USB/SOF_1KHZ | CCL0/IN[4] | - | - | VDDIO | Hi-Z |
| 39 | PD02 | - | - | - | - | - | - | - | USB/DM | - | - | - | VDDIO | Hi-Z |
| 40 | PD03 | - | - | - | - | - | - | - | USB/DP | - | - | - | VDDIO | Hi-Z |
| 41 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 42 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 43 | PD06 | USERLDO | - | - | - | - | SERCOM2/PAD[1] | - | CAN0/RX | CCL1/IN[3] | - | XY[0] | VDDIO | Hi-Z |
| 44 | PD07 | - | EXTINT[4] | - | - | - | SERCOM2/PAD[2] | TCC4/WO[0] | - | CCL1/IN[4] | - | XY[1] | VDDIO | Hi-Z |
| 45 | PD04 | OSCCTRL/XOUT | EXTINT[2] | - | - | - | - | - | - | - | IO[1] | - | VDDIO | Hi-Z |
| 46 | PD05 | OSCCTRL/XIN | EXTINT[3] | - | - | - | - | - | - | - | IO[0] | - | VDDIO | Hi-Z |
| 47 | PD08 | - | EXTINT[5] | - | - | - | SERCOM2/PAD[3] | TCC4/WO[1] | CAN0/TX | CCL0/OUT[1] | - | XY[2] | VDDIO | Hi-Z |
| 48 | PD09 | - | EXTINT[6] | - | - | - | SERCOM3/PAD[0] | TCC5/WO[0] | - | CCL1/OUT[1] | - | XY[3] | VDDIO | Hi-Z |


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

## 5.2. 64-Pin TQFP and 64-Pin VQFN

| 64-Pin VQFN and 64-Pin TQFP (Top View )[1] | | |
|---|---|---|
| PIC32CM5112GC00064 | TQFP |  |
| | VQFN |  |

**Note:**

1. The VQFN package has wettable flanks.


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

**Table 5-2.** 64-Pin VQFN and 64-Pin TQFP

| Pin | MUXEN = 1 | Priority Function | MUXEN = 1 | | | | | | | | | | Supply | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PMUX Values | | | | | | | | | | | |
| | | | 0x0 | 0x1 | | | 0x3 | 0x5 | 0x7 | 0x8 | 0xA | 0xF | | |
| | | | EIC | AC | ADC | ADCN | SERCOM | TCC | CAN | CCL | GCLK | PTC | | |
| 1 | PA00 | - | EXTINT[0] | - | - | - | SERCOM3/PAD[1] | TCC5/WO[1] | CAN1/RX | CCL0/IN[6] | - | XY[4] | VDDIO | Hi-Z |
| 2 | PA01 | - | EXTINT[1] | CMP[0] | - | - | SERCOM3/PAD[2] | TCC6/WO[0] | CAN1/TX | CCL0/IN[7] | - | XY[5] | VDDIO | Hi-Z |
| 3 | PA02 | - | EXTINT[2] | CMP[1] | - | - | SERCOM3/PAD[3] | TCC6/WO[1] | - | CCL0/OUT[2] | - | XY[6] | VDDIO | Hi-Z |
| 4 | PA10 | - | EXTINT[10] | - | - | - | - | TCC4/WO[0] | CAN0/RX | CCL0/IN[8] | - | XY[22] | VDDIO | Hi-Z |
| 5 | PA11 | - | EXTINT[11] | - | - | - | - | TCC4/WO[1] | CAN0/TX | CCL1/IN[8] | - | XY[23] | VDDIO | Hi-Z |
| 6 | PA12 | - | EXTINT[12] | - | - | - | - | - | - | CCL1/IN[8] | - | XY[24] | VDDIO | Hi-Z |
| 7 | PA03 | - | EXTINT[3] | - | AIN[0] | - | SERCOM4/PAD[0] | - | - | - | - | XY[7] | VDDIO | Hi-Z |
| 8 | PA04 | - | EXTINT[4] | - | AIN[1] | ANN[0] | SERCOM4/PAD[1] | - | - | - | - | XY[8] | VDDIO | Hi-Z |
| 9 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 10 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 11 | PA05 | - | EXTINT[5] | - | AIN[2] | - | SERCOM4/PAD[2] | - | - | CCL1/OUT[2] | - | XY[9] | VDDIO | Hi-Z |
| 12 | PA06 | - | EXTINT[6] | - | AIN[3] | ANN[2] | SERCOM4/PAD[3] | - | - | CCL1/IN[7] | - | XY[10] | VDDIO | Hi-Z |
| 13 | PA13 | - | EXTINT[13] | CMP[0] | - | - | - | - | - | - | - | XY[25] | VDDIO | Hi-Z |
| 14 | PA07 | RTC/IN[0] | EXTINT[7] | AIN[0] | AIN[4] | - | - | - | - | - | - | XY[11] | VDDIO | Hi-Z |
| 15 | PA08 | RTC/IN[1] | EXTINT[8] | AIN[1] | AIN[5] | ANN[4] | - | - | - | - | - | XY[12] | VDDIO | Hi-Z |
| 16 | PA09 | RTC/IN[2] | EXTINT[9] | - | AIN[6] | VREFH | - | - | - | CCL1/IN[6] | - | XY[13] | VDDIO | Hi-Z |
| 17 | PB00 | RTC/IN[3] | EXTINT[0] | AIN[2] | AIN[7] | - | - | - | - | CCL1/OUT[3] | - | XY[14] | VDDIO | Hi-Z |
| 18 | PB07 | RTC/IN[4] | EXTINT[8] | CMP[1] | - | - | - | - | - | CCL1/IN[11] | - | XY[26] | VDDIO | Hi-Z |
| 19 | AVDD | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 20 | AVSS/VSSREG | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 21 | PB01 | XIN32 | EXTINT[1] | - | - | - | - | - | - | - | - | - | VDDIO | Hi-Z |
| 22 | PB02 | XOUT32 | EXTINT[2] | - | - | - | - | - | - | - | - | - | VDDIO | Hi-Z |
| 23 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 24 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 25 | VDDREG | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 26 | PB03 | RTC/OUT[0] | EXTINT[3] | AIN[3] | AIN[8] | - | SERCOM5/PAD[0] | - | - | CCL1/IN[10] | - | XY[15] | VDDIO | Hi-Z |
| 27 | PB04 | RTC/OUT[1], SUPC/OUT[0] | EXTINT[4] | - | AIN[9] | - | SERCOM5/PAD[1] | - | - | CCL1/IN[9] | - | XY[16] | VDDIO | Hi-Z |
| 28 | PB05 | RTC/OUT[2], SUPC/OUT[1] | EXTINT[5] | - | AIN[10] | - | SERCOM5/PAD[2] | - | - | CCL0/IN[9] | - | XY[17] | VDDIO | Hi-Z |
| 29 | PB08 | RTC/IN[5] | EXTINT[9] | - | - | - | - | TCC2/WO[1] | - | CCL1/IN[11] | - | XY[27] | VDDIO | Hi-Z |
| 30 | PB09 | RTC/OUT[4] | EXTINT[10] | - | - | - | - | TCC2/WO[0] | - | - | IO[7] | XY[28] | VDDIO | Hi-Z |
| 31 | PB06 | RTC/OUT[3] | EXTINT[6] | - | AIN[11] | - | SERCOM5/PAD[3] | - | - | CCL0/IN[10] | IO[0] | XY[18] | VDDIO | Hi-Z |
| 32 | PB10 | RTC/OUT[5] | EXTINT[11] | - | - | - | - | - | - | CCL0/IN[11] | IO[6] | XY[29] | VDDIO | Hi-Z |
| 33 | PC00 | - | EXTINT[0] | - | - | - | SERCOM0/PAD[0] | TCC0/WO[0] | - | CCL0/OUT[3] | IO[1] | XY[19] | VDDIO | Hi-Z |
| 34 | PC01 | - | EXTINT[1] | - | - | - | SERCOM0/PAD[1] | TCC0/WO[1] | CAN0/RX | CCL0/IN[0] | IO[2] | XY[20] | VDDIO | Hi-Z |
| 35 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 36 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 37 | PC02 | - | EXTINT[2] | - | - | - | SERCOM0/PAD[2] | TCC1/WO[0] | CAN0/TX | CCL0/IN[1] | IO[3] | XY[21] | VDDIO | Hi-Z |
| 38 | PC03 | - | EXTINT[3] | - | - | - | SERCOM0/PAD[3] | TCC1/WO[1] | - | CCL0/OUT[0] | IO[4] | ECI[0] | VDDIO | Hi-Z |
| 39 | PC09 | - | EXTINT[9] | - | - | - | - | TCC1/WO[0] | - | CCL0/IN[2] | IO[0] | XY[30] | VDDIO | Hi-Z |
| 40 | PC04 | - | EXTINT[4] | - | - | - | SERCOM2/PAD[0] | TCC2/WO[0] | - | CCL1/IN[0] | IO[5] | ECI[1] | VDDIO | Hi-Z |
| 41 | PC10 | - | EXTINT[10] | - | - | - | - | TCC1/WO[1] | - | CCL1/IN[2] | IO[1] | XY[31] | VDDIO | Hi-Z |
| 42 | $\overline{\text{RESET}}$ | - | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 43 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 44 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 45 | PC05 | TDO | EXTINT[5] | - | - | - | SERCOM1/PAD[3] | TCC2/WO[1] | CAN1/RX | CCL1/IN[1] | IO[1] | - | VDDIO | Hi-Z |
| 46 | PC06 | TDI | EXTINT[6] | - | - | - | SERCOM1/PAD[2] | - | CAN1/TX | CCL1/OUT[0] | - | - | VDDIO | Hi-Z |
| 47 | - | TMS/SWDIO | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 48 | - | TCK/SWCLK | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 49 | PD00 | - | NMI | - | - | - | SERCOM1/PAD[1] | TCC3/WO[0] | - | CCL0/IN[3] | - | - | VDDIO | Hi-Z |
| 50 | PD10 | - | EXTINT[8] | - | - | - | SERCOM1/PAD[2] | TCC5/WO[0] | - | CCL0/IN[5] | - | - | VDDIO | Hi-Z |
| 51 | PD11 | - | EXTINT[9] | - | - | - | SERCOM1/PAD[3] | TCC5/WO[1] | - | CCL1/IN[5] | - | - | VDDIO | Hi-Z |
| 52 | PD01 | - | EXTINT[1] | - | - | - | SERCOM1/PAD[0] | TCC3/WO[1] | USB/SOF_1KHZ | CCL0/IN[4] | - | - | VDDIO | Hi-Z |
| 53 | PD02 | - | - | - | - | - | - | - | USB/DM | - | - | - | VDDIO | Hi-Z |
| 54 | PD03 | - | - | - | - | - | - | - | USB/DP | - | - | - | VDDIO | Hi-Z |
| 55 | PD12 | - | EXTINT[10] | - | - | - | - | TCC6/WO[0] | CAN1/RX | CCL1/IN[5] | - | - | VDDIO | Hi-Z |
| 56 | PD13 | - | EXTINT[11] | - | - | - | - | TCC6/WO[1] | CAN1/TX | - | - | - | VDDIO | Hi-Z |
| 57 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 58 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 59 | PD06 | USERLDO | - | - | - | - | SERCOM2/PAD[1] | - | CAN0/RX | CCL1/IN[3] | - | XY[0] | VDDIO | Hi-Z |


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

**Table 5-2.** 64-Pin VQFN and 64-Pin TQFP (continued)

| Pin | MUXEN = 1 | Priority Function | MUXEN = 1 | | | | | | | | | | Supply | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PMUX Values | | | | | | | | | | | |
| | | | 0x0 | 0x1 | | | 0x3 | 0x5 | 0x7 | 0x8 | 0xA | 0xF | | |
| | | | EIC | AC | ADC | ADCN | SERCOM | TCC | CAN | CCL | GCLK | PTC | | |
| 60 | PD07 | - | EXTINT[4] | - | - | - | SERCOM2/PAD[2] | TCC4/WO[0] | - | CCL1/IN[4] | - | XY[1] | VDDIO | Hi-Z |
| 61 | PD04 | OSCCTRL/XOUT | EXTINT[2] | - | - | - | - | - | - | - | IO[1] | - | VDDIO | Hi-Z |
| 62 | PD05 | OSCCTRL/XIN | EXTINT[3] | - | - | - | - | - | - | - | IO[0] | - | VDDIO | Hi-Z |
| 63 | PD08 | - | EXTINT[5] | - | - | - | SERCOM2/PAD[3] | TCC4/WO[1] | CAN0/TX | CCL0/OUT[1] | - | XY[2] | VDDIO | Hi-Z |
| 64 | PD09 | - | EXTINT[6] | - | - | - | SERCOM3/PAD[0] | TCC5/WO[0] | - | CCL1/OUT[1] | - | XY[3] | VDDIO | Hi-Z |


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

## 5.3. 100-Pin TQFP

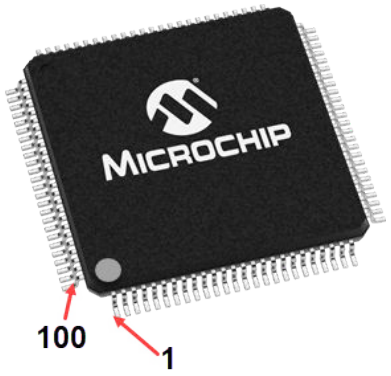
| 100-Pin TQFP (Top View) | |
|---|---|
| PIC32CM5112GC00100 |  |



**Table 5-3.** 100-Pin TQFP

| Pin | MUXEN = 0 | Priority Function | MUXEN = 1 | | | | | | | | | | Supply | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PMUX Values | | | | | | | | | | | |
| | | | 0x0 | 0x1 | | | 0x3 | 0x5 | 0x7 | 0x8 | 0xA | 0xF | | |
| | | | EIC | AC | ADC | ADCN | SERCOM | TCC | CAN | CCL | GCLK | PTC | | |
| 1 | PA00 | - | EXTINT[0] | - | - | - | SERCOM3/PAD[1] | TCC5/WO[1] | CAN1/RX | CCL0/IN[6] | - | XY[4] | VDDIO | Hi-Z |
| 2 | PA14 | - | EXTINT[14] | - | - | - | SERCOM4/PAD[3] | - | - | CCL0/IN[6] | - | XY[32] | VDDIO | Hi-Z |
| 3 | PA01 | - | EXTINT[1] | CMP[0] | - | - | SERCOM3/PAD[2] | TCC6/WO[0] | CAN1/TX | CCL0/IN[7] | - | XY[5] | VDDIO | Hi-Z |
| 4 | PA15 | - | EXTINT[15] | - | - | - | SERCOM5/PAD[3] | - | - | CCL0/IN[7] | - | XY[33] | VDDIO | Hi-Z |
| 5 | PA02 | - | EXTINT[2] | CMP[1] | - | - | SERCOM3/PAD[3] | TCC6/WO[1] | - | CCL0/OUT[2] | - | XY[6] | VDDIO | Hi-Z |
| 6 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 7 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 8 | PA10 | - | EXTINT[10] | - | - | - | - | TCC4/WO[0] | CAN0/RX | CCL0/IN[8] | - | XY[22] | VDDIO | Hi-Z |
| 9 | PA11 | - | EXTINT[11] | - | - | - | - | TCC4/WO[1] | CAN0/TX | CCL1/IN[8] | - | XY[23] | VDDIO | Hi-Z |
| 10 | PA12 | - | EXTINT[12] | - | - | - | - | - | - | CCL1/IN[8] | - | XY[24] | VDDIO | Hi-Z |
| 11 | PA16 | - | EXTINT[0] | - | - | - | SERCOM5/PAD[0] | - | - | CCL1/IN[6] | - | XY[34] | VDDIO | Hi-Z |
| 12 | PA17 | - | EXTINT[1] | - | - | - | SERCOM5/PAD[1] | - | - | CCL1/IN[7] | - | XY[35] | VDDIO | Hi-Z |
| 13 | PA18 | - | EXTINT[2] | - | - | - | SERCOM5/PAD[2] | - | - | CCL0/IN[8] | - | - | VDDIO | Hi-Z |
| 14 | PA03 | - | EXTINT[3] | - | AIN[0] | - | SERCOM4/PAD[0] | - | - | - | - | XY[7] | VDDIO | Hi-Z |
| 15 | PA04 | - | EXTINT[4] | - | AIN[1] | ANN[0] | SERCOM4/PAD[1] | - | - | - | - | XY[8] | VDDIO | Hi-Z |
| 16 | PA05 | - | EXTINT[5] | - | AIN[2] | - | SERCOM4/PAD[2] | - | - | CCL1/OUT[2] | - | XY[9] | VDDIO | Hi-Z |
| 17 | PA06 | - | EXTINT[6] | - | AIN[3] | ANN[2] | SERCOM4/PAD[3] | - | - | CCL1/IN[7] | - | XY[10] | VDDIO | Hi-Z |
| 18 | PA13 | - | EXTINT[13] | CMP[0] | - | - | - | - | - | - | - | XY[25] | VDDIO | Hi-Z |
| 19 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 20 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 21 | PA07 | RTC/IN[0] | EXTINT[7] | AIN[0] | AIN[4] | - | - | - | - | - | - | XY[11] | VDDIO | Hi-Z |
| 22 | PA08 | RTC/IN[1] | EXTINT[8] | AIN[1] | AIN[5] | ANN[4] | - | - | - | - | - | XY[12] | VDDIO | Hi-Z |
| 23 | PA09 | RTC/IN[2] | EXTINT[9] | - | AIN[6] | VREFH | - | - | - | CCL1/IN[6] | - | XY[13] | VDDIO | Hi-Z |
| 24 | PA20 | - | EXTINT[4] | - | - | - | - | TCC0/WO[1] | - | CCL1/OUT[2] | - | - | VDDIO | Hi-Z |
| 25 | PA19 | - | EXTINT[3] | - | - | - | - | TCC0/WO[0] | - | CCL0/OUT[2] | - | - | VDDIO | Hi-Z |
| 26 | PB00 | RTC/IN[3] | EXTINT[0] | AIN[2] | AIN[7] | - | - | - | - | CCL1/OUT[3] | - | XY[14] | VDDIO | Hi-Z |
| 27 | PB07 | RTC/IN[4] | EXTINT[8] | CMP[1] | - | - | - | - | - | CCL1/IN[11] | - | XY[26] | VDDIO | Hi-Z |
| 28 | AVDD | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 29 | AVSS/VSSREG | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 30 | PB01 | XIN32 | EXTINT[1] | - | - | - | - | - | - | - | - | - | VDDIO | Hi-Z |
| 31 | PB02 | XOUT32 | EXTINT[2] | - | - | - | - | - | - | - | - | - | VDDIO | Hi-Z |
| 32 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 33 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 34 | VDDREG | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 35 | PB11 | RTC/IN[6] | EXTINT[12] | - | - | - | - | TCC0/WO[0] | - | CCL1/IN[9] | - | - | VDDIO | Hi-Z |
| 36 | PB12 | RTC/IN[7] | EXTINT[2] | - | - | - | - | TCC0/WO[1] | - | CCL0/IN[9] | - | - | VDDIO | Hi-Z |
| 37 | PB13 | RTC/OUT[6] | EXTINT[14] | - | - | - | - | TCC3/WO[0] | - | CCL1/IN[10] | - | - | VDDIO | Hi-Z |
| 38 | PB14 | RTC/OUT[7] | EXTINT[15] | - | - | - | - | TCC3/WO[1] | - | CCL0/IN[10] | - | - | VDDIO | Hi-Z |
| 39 | PB03 | RTC/OUT[0] | EXTINT[3] | AIN[3] | AIN[8] | - | SERCOM5/PAD[0] | - | - | CCL1/IN[10] | - | XY[15] | VDDIO | Hi-Z |
| 40 | PB04 | RTC/OUT[1], SUPC/OUT[0] | EXTINT[4] | - | AIN[9] | - | SERCOM5/PAD[1] | - | - | CCL1/IN[9] | - | XY[16] | VDDIO | Hi-Z |


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

**Table 5-3.** 100-Pin TQFP (continued)

| Pin | MUXEN = 0 | Priority Function | MUXEN = 1 | | | | | | | | | | Supply | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PMUX Values | | | | | | | | | | | |
| | | | 0x0 | 0x1 | | | 0x3 | 0x5 | 0x7 | 0x8 | 0xA | 0xF | | |
| | | | EIC | AC | ADC | ADCN | SERCOM | TCC | CAN | CCL | GCLK | PTC | | |
| 41 | PB05 | RTC/OUT[2], SUPC/OUT[1] | EXTINT[5] | - | AIN[10] | - | SERCOM5/PAD[2] | - | - | CCL0/IN[9] | - | XY[17] | VDDIO | Hi-Z |
| 42 | PB08 | RTC/IN[5] | EXTINT[9] | - | - | - | - | TCC2/WO[1] | - | CCL1/IN[11] | - | XY[27] | VDDIO | Hi-Z |
| 43 | PB09 | RTC/OUT[4] | EXTINT[10] | - | - | - | - | TCC2/WO[0] | - | - | IO[7] | XY[28] | VDDIO | Hi-Z |
| 44 | PB06 | RTC/OUT[3] | EXTINT[6] | - | AIN[11] | - | SERCOM5/PAD[3] | - | - | CCL0/IN[10] | IO[0] | XY[18] | VDDIO | Hi-Z |
| 45 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 46 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 47 | PB15 | - | EXTINT[0] | - | - | - | SERCOM0/PAD[0] | - | - | CCL0/IN[11] | IO[0] | - | VDDIO | Hi-Z |
| 48 | PB16 | - | EXTINT[1] | - | - | - | SERCOM0/PAD[1] | - | - | CCL0/OUT[3] | IO[6] | - | VDDIO | Hi-Z |
| 49 | PB17 | - | EXTINT[2] | - | - | - | SERCOM0/PAD[2] | - | - | CCL1/OUT[3] | - | - | VDDIO | Hi-Z |
| 50 | PB10 | RTC/OUT[5] | EXTINT[11] | - | - | - | - | - | - | CCL0/IN[11] | IO[6] | XY[29] | VDDIO | Hi-Z |
| 51 | PC00 | - | EXTINT[0] | - | - | - | SERCOM0/PAD[0] | TCC0/WO[0] | - | CCL0/OUT[3] | IO[1] | XY[19] | VDDIO | Hi-Z |
| 52 | PC11 | - | EXTINT[11] | - | - | - | SERCOM0/PAD[3] | - | - | CCL1/IN[0] | - | - | VDDIO | Hi-Z |
| 53 | PC12 | - | EXTINT[12] | - | - | - | SERCOM1/PAD[3] | - | - | CCL0/IN[0] | - | - | VDDIO | Hi-Z |
| 54 | PC01 | - | EXTINT[1] | - | - | - | SERCOM0/PAD[1] | TCC0/WO[1] | CAN0/RX | CCL0/IN[0] | IO[2] | XY[20] | VDDIO | Hi-Z |
| 55 | PC13 | - | EXTINT[13] | - | - | - | SERCOM1/PAD[2] | - | - | CCL0/OUT[0] | - | - | VDDIO | Hi-Z |
| 56 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 57 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 58 | PC02 | - | EXTINT[2] | - | - | - | SERCOM0/PAD[2] | TCC1/WO[0] | CAN0/TX | CCL0/IN[1] | IO[3] | XY[21] | VDDIO | Hi-Z |
| 59 | PC14 | - | EXTINT[14] | - | - | - | SERCOM1/PAD[1] | - | - | CCL0/IN[1] | - | - | VDDIO | Hi-Z |
| 60 | PC15 | - | EXTINT[15] | - | - | - | SERCOM1/PAD[0] | - | - | CCL0/IN[2] | - | - | VDDIO | Hi-Z |
| 61 | PC03 | - | EXTINT[3] | - | - | - | SERCOM0/PAD[3] | TCC1/WO[1] | - | CCL0/OUT[0] | IO[4] | ECI[0] | VDDIO | Hi-Z |
| 62 | PC16 | - | EXTINT[0] | - | - | - | SERCOM2/PAD[3] | - | - | CCL1/IN[2] | - | - | VDDIO | Hi-Z |
| 63 | PC09 | - | EXTINT[9] | - | - | - | - | TCC1/WO[0] | - | CCL0/IN[2] | IO[0] | XY[30] | VDDIO | Hi-Z |
| 64 | PC04 | - | EXTINT[4] | - | - | - | SERCOM2/PAD[0] | TCC2/WO[0] | - | CCL1/IN[0] | IO[5] | ECI[1] | VDDIO | Hi-Z |
| 65 | PC17 | - | EXTINT[1] | - | - | - | SERCOM2/PAD[2] | - | - | CCL1/IN[1] | - | - | VDDIO | Hi-Z |
| 66 | PC10 | - | EXTINT[10] | - | - | - | - | TCC1/WO[1] | - | CCL1/IN[2] | IO[1] | XY[31] | VDDIO | Hi-Z |
| 67 | $\overline{\text{RESET}}$ | - | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 68 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 69 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 70 | PC18 | - | EXTINT[2] | - | - | - | SERCOM2/PAD[1] | - | - | - | - | - | VDDIO | Hi-Z |
| 71 | PC19 | - | EXTINT[3] | - | - | - | SERCOM2/PAD[0] | - | - | CCL1/OUT[0] | - | - | VDDIO | Hi-Z |
| 72 | PC05 | TDO | EXTINT[5] | - | - | - | SERCOM1/PAD[3] | TCC2/WO[1] | CAN1/RX | CCL1/IN[1] | IO[1] | - | VDDIO | Hi-Z |
| 73 | PC06 | TDI | EXTINT[6] | - | - | - | SERCOM1/PAD[2] | - | CAN1/TX | CCL1/OUT[0] | - | - | VDDIO | Hi-Z |
| 74 | - | TMS/SWDIO | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 75 | - | TCK/SWCLK | - | - | - | - | - | - | - | - | - | - | VDDIO | I - PU |
| 76 | PD00 | - | NMI | - | - | - | SERCOM1/PAD[1] | TCC3/WO[0] | - | CCL0/IN[3] | - | - | VDDIO | Hi-Z |
| 77 | PD10 | - | EXTINT[8] | - | - | - | SERCOM1/PAD[2] | TCC5/WO[0] | - | CCL0/IN[5] | - | - | VDDIO | Hi-Z |
| 78 | PD14 | - | EXTINT[12] | - | - | - | SERCOM3/PAD[3] | - | - | CCL0/OUT[1] | - | - | VDDIO | Hi-Z |
| 79 | PD11 | - | EXTINT[9] | - | - | - | SERCOM1/PAD[3] | TCC5/WO[1] | - | CCL1/IN[5] | - | - | VDDIO | Hi-Z |
| 80 | PD01 | - | EXTINT[1] | - | - | - | SERCOM1/PAD[0] | TCC3/WO[1] | USB/SOF_1KHZ | CCL0/IN[4] | - | - | VDDIO | Hi-Z |
| 81 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 82 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 83 | PD02 | - | - | - | - | - | - | - | USB/DM | - | - | - | VDDIO | Hi-Z |
| 84 | PD03 | - | - | - | - | - | - | - | USB/DP | - | - | - | VDDIO | Hi-Z |
| 85 | PD15 | - | EXTINT[13] | - | - | - | SERCOM3/PAD[1] | - | - | CCL1/OUT[1] | - | - | VDDIO | Hi-Z |
| 86 | PD16 | - | EXTINT[14] | - | - | - | SERCOM3/PAD[0] | - | - | CCL1/IN[4] | - | - | VDDIO | Hi-Z |
| 87 | PD17 | - | EXTINT[15] | - | - | - | SERCOM3/PAD[2] | - | - | CCL0/IN[4] | - | - | VDDIO | Hi-Z |
| 88 | PD12 | - | EXTINT[10] | - | - | - | - | TCC6/WO[0] | CAN1/RX | CCL1/IN[5] | - | - | VDDIO | Hi-Z |
| 89 | PD18 | - | EXTINT[0] | - | - | - | SERCOM4/PAD[2] | - | - | CCL0/IN[3] | - | - | VDDIO | Hi-Z |
| 90 | PD13 | - | EXTINT[11] | - | - | - | - | TCC6/WO[1] | CAN1/TX | - | - | - | VDDIO | Hi-Z |
| 91 | VSS | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 92 | VDDIO | - | - | - | - | - | - | - | - | - | - | - | - | - |
| 93 | PD06 | USERLDO | - | - | - | - | SERCOM2/PAD[1] | - | CAN0/RX | CCL1/IN[3] | - | XY[0] | VDDIO | Hi-Z |
| 94 | PD07 | - | EXTINT[4] | - | - | - | SERCOM2/PAD[2] | TCC4/WO[0] | - | CCL1/IN[4] | - | XY[1] | VDDIO | Hi-Z |
| 95 | PD19 | - | EXTINT[1] | - | - | - | SERCOM4/PAD[0] | - | - | CCL0/IN[5] | - | - | VDDIO | Hi-Z |
| 96 | PD04 | OSCCTRL/XOUT | EXTINT[2] | - | - | - | - | - | - | - | IO[1] | - | VDDIO | Hi-Z |
| 97 | PD05 | OSCCTRL/XIN | EXTINT[3] | - | - | - | - | - | - | - | IO[0] | - | VDDIO | Hi-Z |
| 98 | PD08 | - | EXTINT[5] | - | - | - | SERCOM2/PAD[3] | TCC4/WO[1] | CAN0/TX | CCL0/OUT[1] | - | XY[2] | VDDIO | Hi-Z |
| 99 | PD20 | - | EXTINT[2] | - | - | - | SERCOM4/PAD[1] | - | - | CCL1/IN[3] | - | - | VDDIO | Hi-Z |


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

**Table 5-3.** 100-Pin TQFP (continued)

| Pin | MUXEN = 0 | Priority Function | MUXEN = 1 | | | | | | | | | | Supply | Reset State |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | PMUX Values | | | | | | | | | | | |
| | | | 0x0 | 0x1 | | | 0x3 | 0x5 | 0x7 | 0x8 | 0xA | 0xF | | |
| | | | EIC | AC | ADC | ADCN | SERCOM | TCC | CAN | CCL | GCLK | PTC | | |
| 100 | PD09 | - | EXTINT[6] | - | - | - | SERCOM3/PAD[0] | TCC5/WO[0] | - | CCL1/OUT[1] | - | XY[3] | VDDIO | Hi-Z |


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

# 6. Signal Description

**Table 6-1.** AC Signal Descriptions

| PIN NAME | PIN NUMBER | | | I/O Set | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | | |
| AC_AIN0 | 21 | 14 | 10 | — | I | Analog | Comparator 0 & 1 external analog inputs |
| AC_AIN1 | 22 | 15 | 11 | — | I | Analog | |
| AC_AIN2 | 26 | 17 | 13 | — | I | Analog | |
| AC_AIN3 | 39 | 26 | 21 | — | I | Analog | |
| AC_CMP0 | 3 | 2 | 2 | 1 | O | Analog | Comparator 0 Output |
| AC_CMP0 | 18 | 13 | — | 2 | O | Analog | |
| AC_CMP1 | 5 | 3 | 3 | 1 | O | Analog | Comparator 1 Output |
| AC_CMP1 | 27 | 18 | — | 2 | O | Analog | |

**Table 6-2.** ADC Signal Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| ADC_VINP0 | 14 | 7 | 4 | I | Analog | ADC Module 0 analog inputs |
| ADC_VINP1 | 15 | 8 | 5 | I | Analog | |
| ADC_VINP2 | 16 | 11 | 8 | I | Analog | |
| ADC_VINP3 | 17 | 12 | 9 | I | Analog | |
| ADC_VINP4 | 21 | 14 | 10 | I | Analog | |
| ADC_VINP5 | 22 | 15 | 11 | I | Analog | |
| ADC_VINP6 | 23 | 16 | 12 | I | Analog | |
| ADC_VINP7 | 26 | 17 | 13 | I | Analog | |
| ADC_VINP8 | 39 | 26 | 21 | I | Analog | |
| ADC_VINP9 | 40 | 27 | 22 | I | Analog | |
| ADC_VINP10 | 41 | 28 | 23 | I | Analog | |
| ADC_VINP11 | 44 | 31 | 24 | I | Analog | |
| ADC_VINM0 | 15 | 8 | 5 | I | Analog | |
| ADC_VINM2 | 17 | 12 | 9 | I | Analog | |
| ADC_VINM4 | 22 | 15 | 11 | I | Analog | |
| ADC_VREFH | 23 | 16 | 12 | I | Analog | |

**Table 6-3.** Programming-Debugging Signal Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| TCK_SWCLK | 75 | 48 | 36 | I | ST | Debug Serial Wire Input Clock |
| TDI | 73 | 46 | 34 | I | ST | Debug/Scan JTAG Data Input |
| TDO | 72 | 45 | 33 | O | CMOS | Debug/Scan JTAG Data Out |
| TMS_SWDIO | 74 | 47 | 35 | O | CMOS | Debug Serial Wire Data In/Out |
| RESET | 67 | 42 | 30 | I | ST | MCU Reset input (Active Low) |


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

**Table 6-4.** CAN Signal Descriptions

| PIN NAME | PIN NUMBER | | | I/O Set | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | | |
| CAN0_RX | 8 | 4 | — | 3 | I | ST | CAN0_FD Receive |
| CAN0_RX | 54 | 34 | 26 | 1 | | | |
| CAN0_RX | 93 | 59 | 43 | 2 | | | |
| CAN0_TX | 9 | 5 | — | 3 | O | CMOS | CAN0_FD Transmit |
| CAN0_TX | 58 | 37 | 27 | 1 | | | |
| CAN0_TX | 98 | 63 | 47 | 2 | | | |
| CAN1_RX | 1 | 1 | 1 | 2 | I | ST | CAN1_FD Receive |
| CAN1_RX | 72 | 45 | 33 | 1 | | | |
| CAN1_RX | 88 | 55 | — | 3 | | | |
| CAN1_TX | 3 | 2 | 2 | 2 | O | CMOS | CAN1_FD Transmit |
| CAN1_TX | 73 | 46 | 34 | 1 | | | |
| CAN1_TX | 90 | 56 | — | 3 | | | |

**Table 6-5.** CCL Signal Descriptions

| PIN NAME | PIN NUMBER | | | IO Set | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | | |
| CCL0_IN0 | 53 | — | — | 2 | I | Digital Input | CCL0 Lookup Table Input 0 |
| CCL0_IN0 | 54 | 34 | 26 | 1 | | | |
| CCL0_IN1 | 58 | 37 | 27 | 1 | I | Digital Input | CCL0 Lookup Table Input 1 |
| CCL0_IN1 | 59 | — | — | 2 | | | |
| CCL0_IN10 | 38 | — | — | 2 | I | Digital Input | CCL0 Lookup Table Input 2 |
| CCL0_IN10 | 44 | 31 | 24 | 1 | | | |
| CCL0_IN11 | 47 | — | — | 1 | I | Digital Input | CCL0 Lookup Table Input 3 |
| CCL0_IN11 | 50 | 32 | — | 2 | | | |
| CCL0_IN2 | 60 | — | — | 1 | I | Digital Input | CCL0 Lookup Table Input 4 |
| CCL0_IN2 | 63 | 39 | — | 2 | | | |
| CCL0_IN3 | 76 | 49 | 37 | 1 | I | Digital Input | CCL0 Lookup Table Input 5 |
| CCL0_IN3 | 89 | — | — | 2 | | | |
| CCL0_IN4 | 80 | 52 | 38 | 1 | I | Digital Input | CCL0 Lookup Table Input 6 |
| CCL0_IN4 | 87 | — | — | 2 | | | |
| CCL0_IN5 | 77 | 50 | — | 1 | I | Digital Input | CCL0 Lookup Table Input 7 |
| CCL0_IN5 | 95 | — | — | 2 | | | |
| CCL0_IN6 | 1 | 1 | 1 | 1 | I | Digital Input | CCL0 Lookup Table Input 8 |
| CCL0_IN6 | 2 | — | — | 2 | | | |
| CCL0_IN7 | 3 | 2 | 2 | 2 | I | Digital Input | CCL0 Lookup Table Input 9 |
| CCL0_IN7 | 4 | — | — | 1 | | | |
| CCL0_IN8 | 8 | 4 | — | 2 | I | Digital Input | CCL0 Lookup Table Input 10 |
| CCL0_IN8 | 13 | — | — | 1 | | | |
| CCL0_IN9 | 36 | — | — | 2 | I | Digital Input | CCL0 Lookup Table Input 11 |
| CCL0_IN9 | 41 | 28 | 23 | 1 | | | |
| CCL0_OUT0 | 55 | — | — | 2 | O | Digital Output | CCL0 Lookup Table output 0 |
| CCL0_OUT0 | 61 | 38 | 28 | 1 | | | |
| CCL0_OUT1 | 78 | — | — | 2 | O | Digital Output | CCL0 Lookup Table output 1 |
| CCL0_OUT1 | 98 | 63 | 47 | 1 | | | |


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

**Table 6-5.** CCL Signal Descriptions (continued)

| PIN NAME | PIN NUMBER | | | IO Set | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | | |
| CCL0_OUT2 | 5 | 3 | 3 | 1 | O | Digital Output | CCL0 Lookup Table output 2 |
| CCL0_OUT2 | 25 | — | — | 2 | | | |
| CCL0_OUT3 | 48 | — | — | 2 | O | Digital Output | CCL0 Lookup Table output 3 |
| CCL0_OUT3 | 51 | 33 | 25 | 1 | | | |
| CCL1_IN0 | 52 | — | — | 2 | I | Digital Input | CCL1 Lookup Table Input 0 |
| CCL1_IN0 | 64 | 40 | 29 | 1 | | | |
| CCL1_IN1 | 65 | — | — | 2 | I | Digital Input | CCL1 Lookup Table Input 1 |
| CCL1_IN1 | 72 | 45 | 33 | 1 | | | |
| CCL1_IN10 | 37 | — | — | 2 | I | Digital Input | CCL1 Lookup Table Input 2 |
| CCL1_IN10 | 39 | 26 | 21 | 1 | | | |
| CCL1_IN11 | 27 | 18 | — | 1 | I | Digital Input | CCL1 Lookup Table Input 3 |
| CCL1_IN11 | 42 | 29 | — | 2 | | | |
| CCL1_IN2 | 62 | — | — | 2 | I | Digital Input | CCL1 Lookup Table Input 4 |
| CCL1_IN2 | 66 | 41 | — | 1 | | | |
| CCL1_IN3 | 93 | 59 | 43 | 1 | I | Digital Input | CCL1 Lookup Table Input 5 |
| CCL1_IN3 | 99 | — | — | 2 | | | |
| CCL1_IN4 | 86 | — | — | 2 | I | Digital Input | CCL1 Lookup Table Input 6 |
| CCL1_IN4 | 94 | 60 | 44 | 1 | | | |
| CCL1_IN5 | 79 | 51 | — | 2 | I | Digital Input | CCL1 Lookup Table Input 7 |
| CCL1_IN5 | 88 | 55 | — | 1 | | | |
| CCL1_IN6 | 11 | — | — | 1 | I | Digital Input | CCL1 Lookup Table Input 8 |
| CCL1_IN6 | 23 | 16 | 12 | 2 | | | |
| CCL1_IN7 | 12 | — | — | 2 | I | Digital Input | CCL1 Lookup Table Input 9 |
| CCL1_IN7 | 17 | 12 | 9 | 1 | | | |
| CCL1_IN8 | 9 | 5 | — | 2 | I | Digital Input | CCL1 Lookup Table Input 10 |
| CCL1_IN8 | 10 | 6 | — | 1 | | | |
| CCL1_IN9 | 35 | — | — | 2 | I | Digital Input | CCL1 Lookup Table Input 11 |
| CCL1_IN9 | 40 | 27 | 22 | 1 | | | |
| CCL1_OUT0 | 71 | — | — | 2 | O | Digital Output | CCL1 Lookup Table output 0 |
| CCL1_OUT0 | 73 | 46 | 34 | 1 | | | |
| CCL1_OUT1 | 85 | — | — | 2 | O | Digital Output | CCL1 Lookup Table output 1 |
| CCL1_OUT1 | 100 | 64 | 48 | 1 | | | |
| CCL1_OUT2 | 16 | 11 | 8 | 1 | O | Digital Output | CCL1 Lookup Table output 2 |
| CCL1_OUT2 | 24 | — | — | 2 | | | |
| CCL1_OUT3 | 26 | 17 | 13 | 1 | O | Digital Output | CCL1 Lookup Table output 3 |
| CCL1_OUT3 | 49 | — | — | 2 | | | |

**Table 6-6.** EIC Signal Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| EIC_EXTINT0 | 1/11/26/47/51/62/89 | 1/17/33 | 1/13/25 | I | ST | External Interrupt 0 input |
| EIC_EXTINT1 | 3/12/30/48/54/65/80/95 | 2/21/34/52 | 2/16/26/38 | I | ST | External Interrupt 1 input |
| EIC_EXTINT2 | 5/13/31/36/49/58/70/96/99 | 3/22/37/61 | 3/17/27/45 | I | ST | External Interrupt 2 input |
| EIC_EXTINT3 | 14/25/39/61/71/97 | 7/26/38/62 | 4/21/28/46 | I | ST | External Interrupt 3 input |


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

**Table 6-6.** EIC Signal Descriptions (continued)

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| EIC_EXTINT4 | 15/24/40/64/94 | 8/27/40/60 | 5/22/29/44 | I | ST | External Interrupt 4 input |
| EIC_EXTINT5 | 16/41/72/98 | 11/28/45/63 | 8/23/33/47 | I | ST | External Interrupt 5 input |
| EIC_EXTINT6 | 17/44/73/100 | 12/31/46/64 | 9/24/34/48 | I | ST | External Interrupt 6 input |
| EIC_EXTINT7 | 21 | 14 | 10 | I | ST | External Interrupt 7 input |
| EIC_EXTINT8 | 22/27/77 | 15/18/50 | 11 | I | ST | External Interrupt 8 input |
| EIC_EXTINT9 | 23/42/63/79 | 16/29/39/51 | 12 | I | ST | External Interrupt 9 input |
| EIC_EXTINT10 | 8/43/66/88 | 4/30/41/55 | — | I | ST | External Interrupt 10 input |
| EIC_EXTINT11 | 9/50/52/90 | 5/32/56 | — | I | ST | External Interrupt 11 input |
| EIC_EXTINT12 | 10/35/53/78 | 6 | — | I | ST | External Interrupt 12 input |
| EIC_EXTINT13 | 18/55/85 | 13 | — | I | ST | External Interrupt 13 input |
| EIC_EXTINT14 | 2/37/59/86 | — | — | I | ST | External Interrupt 14 input |
| EIC_EXTINT15 | 4/38/60/87 | — | — | I | ST | External Interrupt 15 input |
| EIC_NMI | 76 | 49 | 37 | I | ST | Non-Maskable Interrupt input |

**Table 6-7.** GCLK Signal Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| GCLK_IO0 | 44/47/63/97 | 31/39/62 | 24/46 | I | ST | External GCLK[7:0] I/O pin Input clock source |
| GCLK_IO1 | 51/66/72/96 | 33/41/45/61 | 25/33/45 | I | ST | |
| GCLK_IO2 | 54 | 34 | 26 | I | ST | |
| GCLK_IO3 | 58 | 37 | 27 | I | ST | |
| GCLK_IO4 | 61 | 38 | 28 | I | ST | |
| GCLK_IO5 | 64 | 40 | 29 | I | ST | |
| GCLK_IO6 | 48/50 | 32 | — | I | ST | |
| GCLK_IO7 | 43 | 30 | — | I | ST | |
| XIN32 | 30 | 21 | 16 | I | Analog | 32.768 kHz oscillator input |
| XOUT32 | 31 | 22 | 17 | O | Analog | 32.768 kHz oscillator output |
| XIN | 97 | 62 | 46 | I | Analog | Primary oscillator input |
| XOUT | 96 | 61 | 45 | O | Analog | Primary oscillator output |


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

**Table 6-8.** PORT Signals Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| PA00 | 1 | 1 | 1 | I/O | ST/CMOS | I/O Port A Pins |
| PA01 | 3 | 2 | 2 | I/O | ST/CMOS | |
| PA02 | 5 | 3 | 3 | I/O | ST/CMOS | |
| PA03 | 14 | 7 | 4 | I/O | ST/CMOS | |
| PA04 | 15 | 8 | 5 | I/O | ST/CMOS | |
| PA05 | 16 | 11 | 8 | I/O | ST/CMOS | |
| PA06 | 17 | 12 | 9 | I/O | ST/CMOS | |
| PA07 | 21 | 14 | 10 | I/O | ST/CMOS | |
| PA08 | 22 | 15 | 11 | I/O | ST/CMOS | |
| PA09 | 23 | 16 | 12 | I/O | ST/CMOS | |
| PA10 | 8 | 4 | — | I/O | ST/CMOS | |
| PA11 | 9 | 5 | — | I/O | ST/CMOS | |
| PA12 | 10 | 6 | — | I/O | ST/CMOS | |
| PA13 | 18 | 13 | — | I/O | ST/CMOS | |
| PA14 | 2 | — | — | I/O | ST/CMOS | |
| PA15 | 4 | — | — | I/O | ST/CMOS | |
| PA16 | 11 | — | — | I/O | ST/CMOS | |
| PA17 | 12 | — | — | I/O | ST/CMOS | |
| PA18 | 13 | — | — | I/O | ST/CMOS | |
| PA19 | 25 | — | — | I/O | ST/CMOS | |
| PA20 | 24 | — | — | I/O | ST/CMOS | |
| PB00 | 26 | 17 | 13 | I/O | ST/CMOS | I/O Port B Pins |
| PB01 | 30 | 21 | 16 | I/O | ST/CMOS | |
| PB02 | 31 | 22 | 17 | I/O | ST/CMOS | |
| PB03 | 39 | 26 | 21 | I/O | ST/CMOS | |
| PB04 | 40 | 27 | 22 | I/O | ST/CMOS | |
| PB05 | 41 | 28 | 23 | I/O | ST/CMOS | |
| PB06 | 44 | 31 | 24 | I/O | ST/CMOS | |
| PB07 | 27 | 18 | — | I/O | ST/CMOS | |
| PB08 | 42 | 29 | — | I/O | ST/CMOS | |
| PB09 | 43 | 30 | — | I/O | ST/CMOS | |
| PB10 | 50 | 32 | — | I/O | ST/CMOS | |
| PB11 | 35 | — | — | I/O | ST/CMOS | |
| PB12 | 36 | — | — | I/O | ST/CMOS | |
| PB13 | 37 | — | — | I/O | ST/CMOS | |
| PB14 | 38 | — | — | I/O | ST/CMOS | |
| PB15 | 47 | — | — | I/O | ST/CMOS | |
| PB16 | 48 | — | — | I/O | ST/CMOS | |
| PB17 | 49 | — | — | I/O | ST/CMOS | |


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

**Table 6-8.** PORT Signals Descriptions (continued)

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| PC00 | 51 | 33 | 25 | I/O | ST/CMOS | I/O Port C Pins |
| PC01 | 54 | 34 | 26 | I/O | ST/CMOS | |
| PC02 | 58 | 37 | 27 | I/O | ST/CMOS | |
| PC03 | 61 | 38 | 28 | I/O | ST/CMOS | |
| PC04 | 64 | 40 | 29 | I/O | ST/CMOS | |
| PC05 | 72 | 45 | 33 | I/O | ST/CMOS | |
| PC06 | 73 | 46 | 34 | I/O | ST/CMOS | |
| PC07 | 74 | 47 | 35 | I/O | ST/CMOS | |
| PC08 | 75 | 48 | 36 | I/O | ST/CMOS | |
| PC09 | 63 | 39 | — | I/O | ST/CMOS | |
| PC10 | 66 | 41 | — | I/O | ST/CMOS | |
| PC11 | 52 | — | — | I/O | ST/CMOS | |
| PC12 | 53 | — | — | I/O | ST/CMOS | |
| PC13 | 55 | — | — | I/O | ST/CMOS | |
| PC14 | 59 | — | — | I/O | ST/CMOS | |
| PC15 | 60 | — | — | I/O | ST/CMOS | |
| PC16 | 62 | — | — | I/O | ST/CMOS | |
| PC17 | 65 | — | — | I/O | ST/CMOS | |
| PC18 | 70 | — | — | I/O | ST/CMOS | |
| PC19 | 71 | — | — | I/O | ST/CMOS | |
| PD00 | 76 | 49 | 37 | I/O | ST/CMOS | I/O Port D Pins |
| PD01 | 80 | 52 | 38 | I/O | ST/CMOS | |
| PD02 | 83 | 53 | 39 | I/O | ST/CMOS | |
| PD03 | 84 | 54 | 40 | I/O | ST/CMOS | |
| PD04 | 96 | 61 | 45 | I/O | ST/CMOS | |
| PD05 | 97 | 62 | 46 | I/O | ST/CMOS | |
| PD06 | 93 | 59 | 43 | I/O | ST/CMOS | |
| PD07 | 94 | 60 | 44 | I/O | ST/CMOS | |
| PD08 | 98 | 63 | 47 | I/O | ST/CMOS | |
| PD09 | 100 | 64 | 48 | I/O | ST/CMOS | |
| PD10 | 77 | 50 | — | I/O | ST/CMOS | |
| PD11 | 79 | 51 | — | I/O | ST/CMOS | |
| PD12 | 88 | 55 | — | I/O | ST/CMOS | |
| PD13 | 90 | 56 | — | I/O | ST/CMOS | |
| PD14 | 78 | — | — | I/O | ST/CMOS | |
| PD15 | 85 | — | — | I/O | ST/CMOS | |
| PD16 | 86 | — | — | I/O | ST/CMOS | |
| PD17 | 87 | — | — | I/O | ST/CMOS | |
| PD18 | 89 | — | — | I/O | ST/CMOS | |
| PD19 | 95 | — | — | I/O | ST/CMOS | |
| PD20 | 99 | — | — | I/O | ST/CMOS | |


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

**Table 6-9.** PTC Signal Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| XY0 | 93 | 59 | 43 | I/O | ST/CMOS | Digital Output (X-line)/Analog I/O (Y-line) |
| XY1 | 94 | 60 | 44 | I/O | ST/CMOS | |
| XY2 | 98 | 63 | 47 | I/O | ST/CMOS | |
| XY3 | 100 | 64 | 48 | I/O | ST/CMOS | |
| XY4 | 1 | 1 | 1 | I/O | ST/CMOS | |
| XY5 | 3 | 2 | 2 | I/O | ST/CMOS | |
| XY6 | 5 | 3 | 3 | I/O | ST/CMOS | |
| XY7 | 14 | 7 | 4 | I/O | ST/CMOS | |
| XY8 | 15 | 8 | 5 | I/O | ST/CMOS | |
| XY9 | 16 | 11 | 8 | I/O | ST/CMOS | |
| XY10 | 17 | 12 | 9 | I/O | ST/CMOS | |
| XY11 | 21 | 14 | 10 | I/O | ST/CMOS | |
| XY12 | 22 | 15 | 11 | I/O | ST/CMOS | |
| XY13 | 23 | 16 | 12 | I/O | ST/CMOS | |
| XY14 | 26 | 17 | 13 | I/O | ST/CMOS | |
| XY15 | 39 | 26 | 21 | I/O | ST/CMOS | |
| XY16 | 40 | 27 | 22 | I/O | ST/CMOS | |
| XY17 | 41 | 28 | 23 | I/O | ST/CMOS | |
| XY18 | 44 | 31 | 24 | I/O | ST/CMOS | |
| XY19 | 51 | 33 | 25 | I/O | ST/CMOS | |
| XY20 | 54 | 34 | 26 | I/O | ST/CMOS | |
| XY21 | 58 | 37 | 27 | I/O | ST/CMOS | |
| XY22 | 8 | 4 | — | I/O | ST/CMOS | |
| XY23 | 9 | 5 | — | I/O | ST/CMOS | |
| XY24 | 10 | 6 | — | I/O | ST/CMOS | |
| XY25 | 18 | 13 | — | I/O | ST/CMOS | |
| XY26 | 27 | 18 | — | I/O | ST/CMOS | |
| XY27 | 42 | 29 | — | I/O | ST/CMOS | |
| XY28 | 43 | 30 | — | I/O | ST/CMOS | |
| XY29 | 50 | 32 | — | I/O | ST/CMOS | |
| XY30 | 63 | 39 | — | I/O | ST/CMOS | |
| XY31 | 66 | 41 | — | I/O | ST/CMOS | |
| XY32 | 2 | — | — | I/O | ST/CMOS | |
| XY33 | 4 | — | — | I/O | ST/CMOS | |
| XY34 | 11 | — | — | I/O | ST/CMOS | |
| XY35 | 12 | — | — | I/O | ST/CMOS | |
| ECI0 | 61 | 38 | 28 | I | Analog | External Capacitor Analog Input 0 |
| ECI1 | 64 | 41 | 29 | I | Analog | External Capacitor Analog Input 1 |


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

**Table 6-10.** RTC Signal Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| IN0 | 21 | 14 | 10 | I | ST | RTC security tamper input loop(s) |
| IN1 | 22 | 15 | 11 | I | ST | |
| IN2 | 23 | 16 | 12 | I | ST | |
| IN3 | 26 | 17 | 13 | I | ST | |
| IN4 | 27 | 18 | — | I | ST | |
| IN5 | 42 | 29 | — | I | ST | |
| IN6 | 35 | — | — | I | ST | |
| IN7 | 36 | — | — | I | ST | |
| OUT0 | 39 | 26 | 21 | O | CMOS | RTC security tamper output loop(s) |
| OUT1 | 40 | 27 | 22 | O | CMOS | |
| OUT2 | 41 | 28 | 23 | O | CMOS | |
| OUT3 | 44 | 31 | 24 | O | CMOS | |
| OUT4 | 43 | 30 | — | O | CMOS | |
| OUT5 | 50 | 32 | — | O | CMOS | |
| OUT6 | 37 | — | — | O | CMOS | |
| OUT7 | 38 | — | — | O | CMOS | |

**Table 6-11.** SERCOM Signal Descriptions

| PIN NAME | PIN NUMBER | | | I/O Set | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | | |
| SERCOM0_PAD0 | 47 | — | — | 2 | I/O | ST/CMOS | SERCOM 0 I/O Pads |
| SERCOM0_PAD0 | 51 | 33 | 25 | 1 | I/O | ST/CMOS | |
| SERCOM0_PAD1 | 48 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM0_PAD1 | 54 | 34 | 26 | 1 | I/O | ST/CMOS | |
| SERCOM0_PAD2 | 49 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM0_PAD2 | 58 | 37 | 27 | 1 | I/O | ST/CMOS | |
| SERCOM0_PAD3 | 52 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM0_PAD3 | 61 | 38 | 28 | 1 | I/O | ST/CMOS | |
| SERCOM1_PAD0 | 60/80 | 52 | 38 | 1 | I/O | ST/CMOS | SERCOM 1 I/O Pads |
| SERCOM1_PAD1 | 59/76 | 49 | 37 | 1 | I/O | ST/CMOS | |
| SERCOM1_PAD2 | 55 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM1_PAD2 | 73 | 46 | 34 | 1 | I/O | ST/CMOS | |
| SERCOM1_PAD2 | 77 | 50 | — | 1 | I/O | ST/CMOS | |
| SERCOM1_PAD3 | 53 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM1_PAD3 | 72 | 45 | 33 | 1 | I/O | ST/CMOS | |
| SERCOM1_PAD3 | 79 | 51 | — | 2 | I/O | ST/CMOS | |
| SERCOM2_PAD0 | 64 | 40 | 29 | 1 | I/O | ST/CMOS | SERCOM 2 I/O Pads |
| SERCOM2_PAD0 | 71 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM2_PAD1 | 70 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM2_PAD1 | 93 | 59 | — | 1 | I/O | ST/CMOS | |
| SERCOM2_PAD2 | 65 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM2_PAD2 | 94 | 60 | 44 | 1 | I/O | ST/CMOS | |
| SERCOM2_PAD3 | 62 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM2_PAD3 | 98 | 63 | 47 | 1 | I/O | ST/CMOS | |


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

**Table 6-11.** SERCOM Signal Descriptions (continued)

| PIN NAME | PIN NUMBER | | | I/O Set | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | | |
| SERCOM3_PAD0 | 86 | — | — | 2 | I/O | ST/CMOS | SERCOM 3 I/O Pads |
| SERCOM3_PAD0 | 100 | 64 | 48 | 1 | I/O | ST/CMOS | |
| SERCOM3_PAD1 | 1 | 1 | 1 | 1 | I/O | ST/CMOS | |
| SERCOM3_PAD1 | 85 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM3_PAD2 | 3 | 2 | 2 | 1 | I/O | ST/CMOS | |
| SERCOM3_PAD2 | 87 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM3_PAD3 | 5 | 3 | 3 | 1 | I/O | ST/CMOS | |
| SERCOM3_PAD3 | 78 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM4_PAD0 | 14 | 7 | 4 | 1 | I/O | ST/CMOS | SERCOM 4 I/O Pads |
| SERCOM4_PAD0 | 95 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM4_PAD1 | 15 | 8 | 5 | 1 | I/O | ST/CMOS | |
| SERCOM4_PAD1 | 99 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM4_PAD2 | 16 | 11 | 8 | 1 | I/O | ST/CMOS | |
| SERCOM4_PAD2 | 89 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM4_PAD3 | 2 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM4_PAD3 | 17 | 12 | 9 | 1 | I/O | ST/CMOS | |
| SERCOM5_PAD0 | 11 | — | — | 2 | I/O | ST/CMOS | SERCOM 5 I/O Pads |
| SERCOM5_PAD0 | 39 | 26 | 21 | 1 | I/O | ST/CMOS | |
| SERCOM5_PAD1 | 12 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM5_PAD1 | 40 | 27 | 22 | 1 | I/O | ST/CMOS | |
| SERCOM5_PAD2 | 13 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM5_PAD2 | 41 | 28 | 23 | 1 | I/O | ST/CMOS | |
| SERCOM5_PAD3 | 4 | — | — | 2 | I/O | ST/CMOS | |
| SERCOM5_PAD3 | 44 | 31 | 24 | 1 | I/O | ST/CMOS | |

**Table 6-12.** TCC Signal Descriptions

| PIN NAME | PIN NUMBER | | | I/O SET | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/ VQFN | 48 TQFP/ VQFN | | | | |
| TCC0_WO0 | 25 | — | — | 2 | I/O | ST/CMOS | TCC0 Waveform Outputs/Input Counter/ Capture |
| TCC0_WO0 | 35 | — | — | 3 | I/O | ST/CMOS | |
| TCC0_WO0 | 51 | 33 | 25 | 1 | I/O | ST/CMOS | |
| TCC0_WO1 | 24 | — | — | 2 | I/O | ST/CMOS | |
| TCC0_WO1 | 36 | — | — | 3 | I/O | ST/CMOS | |
| TCC0_WO1 | 54 | 34 | 26 | 1 | I/O | ST/CMOS | |
| TCC1_WO0 | 58 | 37 | 27 | 1 | I/O | ST/CMOS | TCC1 Waveform Outputs/Input Counter/ Capture |
| TCC1_WO0 | 63 | 39 | — | 2 | I/O | ST/CMOS | |
| TCC1_WO1 | 61 | 38 | 28 | 1 | I/O | ST/CMOS | |
| TCC1_WO1 | 66 | 41 | — | 2 | I/O | ST/CMOS | |
| TCC2_WO0 | 43 | 30 | — | 2 | I/O | ST/CMOS | TCC2 Waveform Outputs/Input Counter/ Capture |
| TCC2_WO0 | 64 | 40 | 29 | 1 | I/O | ST/CMOS | |
| TCC2_WO1 | 42 | 29 | — | 2 | I/O | ST/CMOS | |
| TCC2_WO1 | 72 | 45 | 33 | 1 | I/O | ST/CMOS | |


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

**Table 6-12.** TCC Signal Descriptions (continued)

| PIN NAME | PIN NUMBER | | | I/O SET | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/ VQFN | 48 TQFP/ VQFN | | | | |
| TCC3_WO0 | 37 | — | — | 2 | I/O | ST/CMOS | TCC3 Waveform Outputs/Input Counter/ Capture |
| TCC3_WO0 | 76 | 49 | 37 | 1 | I/O | ST/CMOS | |
| TCC3_WO1 | 38 | — | — | 2 | I/O | ST/CMOS | |
| TCC3_WO1 | 80 | 52 | 38 | 1 | I/O | ST/CMOS | |
| TCC4_WO0 | 8 | 4 | — | 2 | I/O | ST/CMOS | TCC4 Waveform Outputs/Input Counter/ Capture |
| TCC4_WO0 | 94 | 60 | 44 | 1 | I/O | ST/CMOS | |
| TCC4_WO1 | 9 | 5 | — | 2 | I/O | ST/CMOS | |
| TCC4_WO1 | 98 | 63 | 47 | 1 | I/O | ST/CMOS | |
| TCC5_WO0 | 77 | 50 | — | 2 | I/O | ST/CMOS | TCC5 Waveform Outputs/Input Counter/ Capture |
| TCC5_WO0 | 100 | 64 | 48 | 1 | I/O | ST/CMOS | |
| TCC5_WO1 | 1 | 1 | 1 | 1 | I/O | ST/CMOS | |
| TCC5_WO1 | 79 | 51 | — | 2 | I/O | ST/CMOS | |
| TCC6_WO0 | 3 | 2 | 2 | 1 | I/O | ST/CMOS | TCC6 Waveform Outputs/Input Counter/ Capture |
| TCC6_WO0 | 88 | 55 | — | 2 | I/O | ST/CMOS | |
| TCC6_WO1 | 5 | 3 | 3 | 1 | I/O | ST/CMOS | |
| TCC6_WO1 | 90 | 56 | — | 2 | I/O | ST/CMOS | |

**Table 6-13.** USB Signal Descriptions

| PIN NAME | PIN NUMBER | | | PIN TYPE | FUNCTION TYPE | DESCRIPTION |
|---|---|---|---|---|---|---|
| | 100 TQFP | 64 TQFP/VQFN | 48 TQFP/VQFN | | | |
| USB_SOF_1KHZ | 80 | 52 | 38 | O | CMOS | USB FS Start of Frame |
| USB_DM | 83 | 53 | 39 | I/O | Analog | USB FS D- differential data line |
| USB_DP | 84 | 54 | 40 | I/O | Analog | USB FS D+ differential data line |


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

# 7. Power Supplies and Startup Considerations

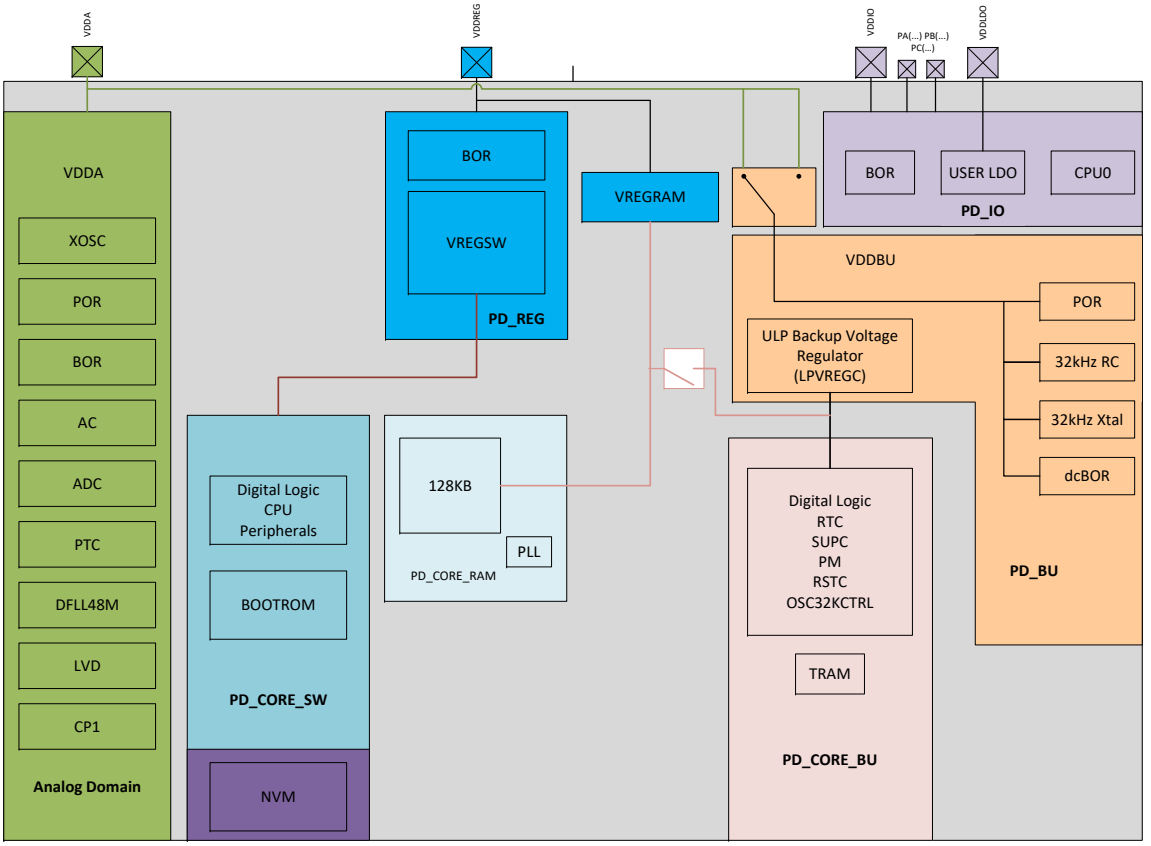
**Table 7-1.** External Power Supply Pins

| Name | Associated Ground | Powers |
|---|---|---|
| AVDD | AVSS | Supplies power to the pSWITCH (in the DS SMOR), SMOR, ADCCOREn, ACDACn, ACCOREn, PTC, PLL, XOSC, DFLL48M and analog I/Os.<br>(Nominal range: 1.71V-3.63V, must be within 300 mv of VDDIO) |
| VDDIO | VSSIO | Powers I/Os.<br>(Nominal range: 1.71V-3.63V, with respect to VSSIO) |
| VDDREG | VSSCORE | Powers all internal regulators except for the LPVREGC.<br>(Nominal range: 1.71V-3.63V, cannot exceed VDDIO) |

**Note:** VDDA and AVDD are used interchangeably throughout this document.

## 7.1. Power Domain Overview

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




## 7.2. Power Domain Constraints

The PIC32CM GC00 power domains are not independent of each other:

- Refer to the *Power, Ground, and Voltage Reference Pinout I/O Descriptions* to understand which pins are tied together.
- VDDIO and VDDREG share VDD. VSSIO and VSSCORE share GND. Ideally, they can be connected together outside of this device.


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

- VDDREG can be used as a low-voltage input to the main regulators, such as 1.8v < VDDREG < AVDD.
- AVDD has its own power and ground, AVSS.
- Ideally, AVDD and VDDIO should be connected together, or have the same voltage.
- VDDREG powers internal regulators that, in turn, power internal domains, such as VDDCORE_SW, VDDCORE_RAM, and VDDCORE_BU.

The maximum supply falling and rising rates of the different power supplies must not exceed the values described in the "Power Supply Characteristics" section of the *Electrical Characteristics (85°C)* chapter.

### 7.2.1. Power Supplies

The PIC32CM GC00 device has the following power supply pins:

- AVDD – Powers the Backup power domain as well as the ADC, Analog Comparator (AC), and the Peripheral Touch Controller (PTC).
- VDDREG – Powers the internal regulators for the VDDCORE_SW, VDDCORE_RAM, and VDDCORE_BU power domains.
- VDDIO – Powers I/O lines, an External Crystal Oscillator (XOSC), the 48 MHz Digital Frequency Locked Loop (DFLL48), and two charge pumps which support the Analog Comparator (AC), 48 MHz Oscillator, and the ADCs on the device (ADC0 and PTC ADC). Most VDDIO pins also power the device's Flash panel through double bonding.

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

### 7.2.2. Power-on Reset (POR) and Brown-out Reset (BOR)

The PIC32CM GC00 embeds three features to monitor, warn, and reset the device:

- A Power-on Reset on $V_{DD}$ (AVDD):
  - Monitoring is always activated, including during device startup or during any sleep modes.
  - Having $V_{DD}$ below a fixed threshold voltage will reset the whole device.

  **Note:** Refer to the *Electrical Characteristics (85°C)* chapter for the rising and falling threshold voltages.
- A Brown-out Reset (VDDBOR) on $V_{DD}$ (VDDREG and VDDIO):
  - The VDDBOR monitors VDD continuously.
  - Duty-Cycled Brown-out Reset (dcBOR) in (continuous mode) or periodically (sampled mode) with a programmable sample frequency in Backup mode.
  - A programmable threshold loaded from the USERCFG memory is used to trigger an interrupt and/or reset the whole device.
- PORCORE a Brown-out Detector on VDDCORE
  **Notes:**
  a. USERCFG (peripheral) entries are defined in the Boot Configuration Section, BOOTCFG1 Table.
  b. PORCORE is calibrated in production and its calibration parameters are stored in the USERCFG memory. This data must not be changed to ensure correct device behavior.

### 7.2.3. Voltage Regulator

The PIC32CM GC00 internal main voltage regulators have three modes, controlled by the Power Manager (PM).

- **Enabled mode:** This is the default mode when CPU and peripherals are running.
- **Low-Power mode:** This is the mode when CPU and peripherals are in Standby, Backup, or Hibernate modes.


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

- **Disabled mode:** When the chip is in Backup mode, the internal regulators (VDDCORE_SW and VDDCORE_RAM) are off, the VDDCORE_SW core power domain is off. The VDDCORE_BU backup domain is powered by the backup regulator (LPVREGC).

### 7.2.4. Internal Power Domains

There are a variety of internal regulated power domains (nominal 1.2v) as shown:

- **VDDCORE_BU**: Powers the backup domain.
- **VDDCORE_SW**: This is the main voltage domain for the CPU, bus, and most peripherals.
- **VDDCORE_RAM**: This domain is used to retain the System SRAM.

### 7.2.5. Typical Powering Schematic

The PIC32CM GC00 can use a single or dual supply from 1.71V to 3.63V.

The following figure shows the recommended power supply connection.

**Figure 7-2.** Power Supply Connection for Single Voltage Mode

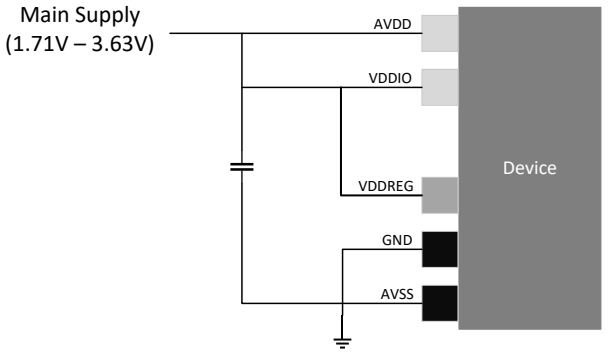
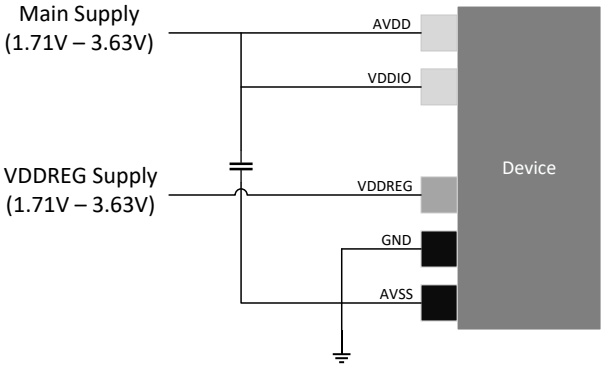



**Figure 7-3.** Power Supply for Dual Voltage Mode




## 7.3. Device Startup

This section summarizes the PIC32CM GC00 device startup sequence which starts after device power-up. After power-up, the device is kept in reset until the power has stabilized throughout the device. Once AVDD/VDDIO/VDDF and VDDCORE (BU, RAM, and SWn) voltages reach a stable value, the internal reset is released.


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

### 7.3.1. Starting of Internal Regulators

After power-up of the VDDREG, VDDIO, and AVDD supplies (VDDx > POR voltage), the device is set to its initial state and kept in reset, until the band-gap and internal regulator power has stabilized throughout the device.

The internal regulators provide power to the VDDCORE_SW and VDDCORE_RAM domains. Once the external voltage VDDIO/AVDD/VDDREG and the internal VDDCORE domains reach a stable value, the internal System Reset is released.

### 7.3.2. Clocks Startup

The device selects the OSC48M oscillator which is enabled by default after reset and configured at 48 MHz.

This 48 MHz clock is also the default time base for the Generic Clock Generator 0 which provides the main clock (CLK_MAIN) to the system through the GLCK_MAIN clock.
**Note:** Other generic clocks are disabled to optimize power consumption.

Some synchronous clocks also need to be active after startup.

**Note:** These active synchronous clocks also receive the 48 MHz clock from Generic Clock Generator 0.

Refer to the *Clock Mask Register* section in the *Main Clock (MCLK)* chapter to obtain the list of clocks that are running by default.

### 7.3.3. I/O Pins

After reset, the I/O pins are tri-stated except the PC08 pin (configured as an input with pull-up enabled) which is by default assigned to the SWCLK peripheral function to allow debugger probe detection.

## 7.4. Analog Peripherals Considerations

This chapter provides a global view of the analog system, which is composed of the following analog peripherals:

- ADC – Analog-to-Digintal Converter
- AC – Analog Comparator (and DAC)
- PTC – Peripheral Touch Controller

The analog peripherals can be connected to each other as illustrated in the block diagrams for the individual analog modules.

**Important:**
When an analog peripheral is enabled, each analog output of the peripheral will be prevented from using the alternative functions of the output pads. This is also true even when the peripheral is used for internal purposes.

Analog inputs do not interfere with alternate pad functions.


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

# 8. Product Mapping

**Table 8-1.** Address Space

| Start Address | End Address | Size (KB) | Name |
|---|---|---|---|
| 0x04000000 | 0x0400FFFF | 64 | ROM |
| 0x04010000 | 0x07FFFFFF | – | Reserved |
| 0x08000000 | 0x08003FFF | 16 | BFM |
| 0x08004000 | 0x09FFFFFF | – | Reserved |
| 0x0A000000 | 0x0A007FFF | 32 | CFM |
| 0x0A008000 | 0x0BFFFFFF | – | Reserved |
| 0x0C000000 | 0x0C07FFFF | 512 | PFM |
| 0x0C080000 | 0x1FFFFFFF | – | Reserved |
| 0x20000000 | 0x2001FFFF | 128 | SRAM |
| 0x20020000 | 0x43FFFFFF | – | Reserved |
| 0x44000000 | 0x44033FFF | 208 | APB0 |
| 0x44034000 | 0x447FFFFF | – | Reserved |
| 0x44800000 | 0x44839FFF | 232 | APB1 |
| 0x4483A000 | 0x44FFFFFF | – | Reserved |
| 0x45000000 | 0x4500FFFF | 64 | APB2 |
| 0x45010000 | 0x4EFFFFFF | – | Reserved |
| 0x4F000000 | 0x4F017FFF | – | Reserved |

**Table 8-2.** CFM

| Start Address | End Address | Size (KB) | Name |
|---|---|---|---|
| 0x0A000000 | 0x0A000FFF | 4 | BOOTCFG1A |
| 0x0A001000 | 0x0A001FFF | 4 | USEROTP |
| 0x0A002000 | 0x0A002FFF | 4 | BOOTCFG1 |
| 0x0A003000 | 0x0A003FFF | 4 | ROMCFG |
| 0x0A004000 | 0x0A004FFF | – | Reserved |
| 0x0A005000 | 0x0A005FFF | – | Reserved |
| 0x0A006000 | 0x0A006FFF | – | Reserved |
| 0x0A007000 | 0x0A007FFF | 4 | CalOTP |

**Table 8-3.** APB0

| Start Address | End Address | Size (KB) | Name |
|---|---|---|---|
| 0x44000000 | 0x44001FFF | 8 | DSU |
| 0x44002000 | 0x44003FFF | 8 | FCR |
| 0x44004000 | 0x44005FFF | 8 | FCW |
| 0x44006000 | 0x44007FFF | 8 | OSCCTRL |
| 0x44008000 | 0x44009FFF | 8 | GCLK |
| 0x4400A000 | 0x4400BFFF | 8 | MCLK |
| 0x4400C000 | 0x4400DFFF | 8 | FREQM |
| 0x4400E000 | 0x4400FFFF | 8 | WDT |
| 0x44010000 | 0x44011FFF | 8 | HMATRIX2 |
| 0x44014000 | 0x4401FFFF | – | Reserved |
| 0x44020000 | 0x44021FFF | 8 | EIC |
| 0x44022000 | 0x44023FFF | 8 | PAC |


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

**Table 8-3.** APB0 (continued)

| Start Address | End Address | Size (KB) | Name |
|---|---|---|---|
| 0x44024000 | 0x44025FFF | – | Reserved |
| 0x44026000 | 0x44027FFF | – | Reserved |
| 0x44028000 | 0x44029FFF | – | Reserved |
| 0x4402A000 | 0x4402BFFF | – | Reserved |
| 0x4402C000 | 0x4402DFFF | 8 | MCRAMC |
| 0x4402E000 | 0x4402FFFF | 8 | CAN0 |
| 0x44030000 | 0x44031FFF | 8 | CAN1 |
| 0x44032000 | 0x44033FFF | 8 | H2PB0 |

**Table 8-4.** APB1

| Start Address | End Address | Size (KB) | Name |
|---|---|---|---|
| 0x44800000 | 0x44801FFF | 8 | PORT |
| 0x44802000 | 0x44803FFF | 8 | DMAC |
| 0x44804000 | 0x44805FFF | 8 | BROMC |
| 0x44806000 | 0x44807FFF | 8 | EVSYS |
| 0x44808000 | 0x44809FFF | 8 | SERCOM0 |
| 0x4480A000 | 0x4480BFFF | 8 | SERCOM1 |
| 0x4480C000 | 0x4480DFFF | 8 | SERCOM2 |
| 0x4480E000 | 0x4480FFFF | 8 | SERCOM3 |
| 0x44810000 | 0x44811FFF | 8 | TCC0 |
| 0x44812000 | 0x44813FFF | 8 | TCC1 |
| 0x44814000 | 0x44815FFF | 8 | TCC2 |
| 0x44816000 | 0x44817FFF | 8 | TCC3 |
| 0x44818000 | 0x44819FFF | 8 | ADC |
| 0x4481A000 | 0x4481FFFF | – | Reserved |
| 0x44820000 | 0x44821FFF | 8 | SERCOM4 |
| 0x44822000 | 0x44823FFF | 8 | SERCOM5 |
| 0x44824000 | 0x44825FFF | 8 | TCC4 |
| 0x44826000 | 0x44827FFF | 8 | TCC5 |
| 0x44828000 | 0x44829FFF | 8 | TCC6 |
| 0x4482A000 | 0x4482BFFF | 8 | AC |
| 0x4482C000 | 0x4482DFFF | 8 | PTC |
| 0x4482E000 | 0x4482FFFF | 8 | CCL0 |
| 0x44830000 | 0x44831FFF | 8 | CCL1 |
| 0x44832000 | 0x44833FFF | 8 | USB |
| 0x44836000 | 0x44837FFF | – | Reserved |
| 0x44838000 | 0x44839FFF | 8 | H2PB1 |

**Table 8-5.** APB2

| Start Address | End Address | Size (KB) | Name |
|---|---|---|---|
| 0x45000000 | 0x45001FFF | 8 | PM |
| 0x45002000 | 0x45003FFF | 8 | SUPC |
| 0x45004000 | 0x45005FFF | 8 | RSTC |
| 0x45006000 | 0x45007FFF | 8 | RTC |
| 0x45008000 | 0x45009FFF | 8 | OSC32KCTRL |
| 0x4500A000 | 0x4500BFFF | 8 | TRAM |


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

**Table 8-5.** APB2 (continued)

| Start Address | End Address | Size (KB) | Name |
|---|---|---|---|
| 0x4500E000 | 0x4500FFFF | 8 | H2PB2 |


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

# 9. Peripherals

## 9.1. Register Description

### 9.1.1. Registers Properties

Registers can be 8, 16, or 32 bits wide. Atomic 8-bit, 16-bit, 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.

#### PAC Write-Protection Register Property

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

#### Read-Synchronized, Write-Synchronized Register Property

Some registers (or bit fields within a register) require synchronization when read and/or written. Synchronization is denoted by the "Read-Synchronized" (bits) and "Write-Synchronized" (bits) property in each individual register description. For additional information, refer to the *Register Synchronization.*

#### Enable-Protected Register Property

Some registers (or bit fields within a register) can only be written when the peripheral is disabled. Such protection is denoted by the "Enable-Protected" (bits) property in each individual register description.

#### Bit field Access Properties

The access properties of bit fields within a register are defined as follows:

**Table 9-1.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |


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

### 9.1.2. Registers Access Permissions

For PIC32CM GC family of devices, there is no notion of secure vs non-secure: all peripherals are accessible.

### 9.1.3. Register Synchronization

#### 9.1.3.1. Overview

All peripherals are composed of one digital bus interface connected to the APB, AHB, or AXI 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, therefore 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.

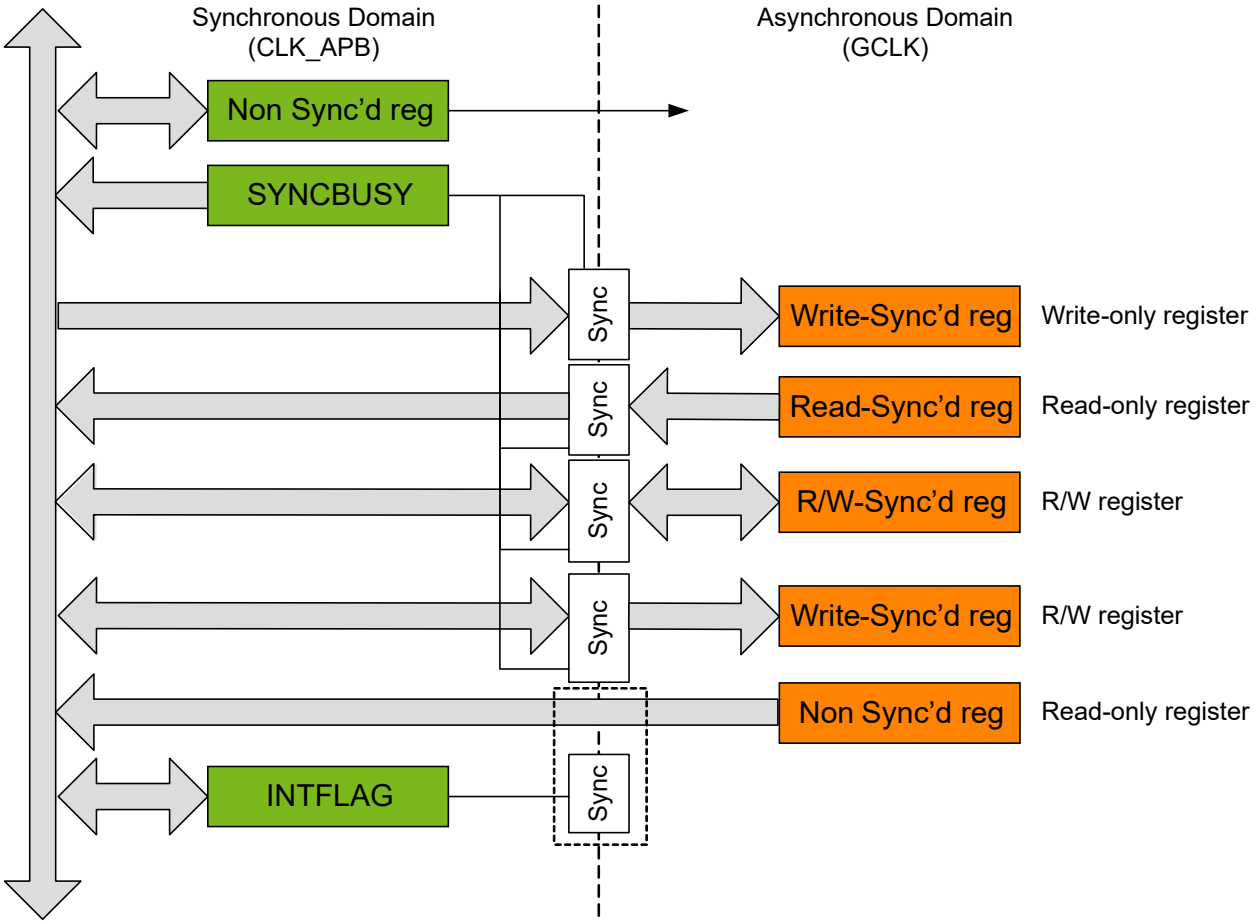
All registers in the peripheral core are synchronized when written. Some registers in the peripheral core are synchronized when read.

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).
**Note:** For registers requiring both read-synchronization and write-synchronization, the corresponding bit in the SYNCBUSY register is shared.


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

**Figure 9-1.** Register Synchronization Overview




### 9.1.3.2. 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. For additional information, refer to the *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).

For example, the REGA and REGB are 8-bit core registers, and the REGC is a 16-bit core register.

| Offset | Register |
|---|---|
| 0x00 | REGA |
| 0x01 | REGB |
| 0x02 | REGC |
| 0x03 | |

Synchronization is per register, hence multiple registers can be synchronized in parallel. Consequently, after the REGA (8-bit access) was written, the REGB (8-bit access) can be written immediately without error.

The REGC (16-bit access) can be written without affecting the REGA or REGB registers. If the REGC register 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. The REGA, REGB, and REGC registers can be updated at different times because of independent write synchronization.

#### 9.1.3.3. 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.
Moreover, if a register is also write-synchronized, any write access while the SYNCBUSY bit is set will be discarded and generate an error.

#### 9.1.3.4. Completion of Synchronization

In order to check if synchronization is complete, the user can either poll the relevant bits in SYNCBUSY or use the Synchronization Ready interrupt (if available). The Synchronization Ready interrupt flag will be set when all ongoing synchronizations are complete, that is, when all bits in SYNCBUSY are '0'.

#### 9.1.3.5. Synchronization Delay

The synchronization will delay read and writer 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 $2 \times P_{APB}$.

### 9.1.4. Interrupt Handling

As a general rule, interrupt flags must be cleared and then read back to confirm they are cleared before exiting the interrupt routine to avoid double interrupts.


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

### 9.1.5. Peripheral Dependencies

**Table 9-2.** Peripherals Configuration Summary

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| **H2PB0 Bridge Peripherals** | | | | | | | | | | | | |
| DSU | 0x44000000 | - | CLMSK0.0 (AHB) | 0 | - | 0 | - | - | - | 1: DCC0<br>2: DCC1 | 0.0<br>- | CORE_SW |
| FCR | 0x44002000 | 0: FCR | CLMSK0.1 (AHB) | 0 | - | 1 | - | - | - | - | 0.1 | CORE_SW |
| FCW | 0x44004000 | 1: FCW | CLMSK0.2 (AHB) | 0 | - | 2 | - | - | - | - | 0.2 | CORE_SW |
| OSCCTRL | 0x44006000 | 4: XOSCRDY<br>4: XOSCFAIL<br>4: CLKFAIL<br>5: DFLLRDY<br>5: DFLLLOCK<br>5: DFLLOVF<br>5: DFLLUNF<br>5: DFLLRCS<br>5: DFLLFAIL<br>6: PLLLOCKR_0<br>6: PLLLOCKF_0 | CLKMSK0.6 (APB) | 0 | 0: DFLL48M<br>1: PLL<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>- | 6 | 2: XOSCFAIL | - | - | - | 0.3 | CORE_SW |
| GCLK | 0x44008000 | - | CLKMSK0.8 (APB) | 0 | - | 8 | - | - | - | - | 0.4 | CORE_SW |
| MCLK | 0x4400A000 | 9: CKRDY | CLKMSK0.9 (APB) | 0 | - | 9 | - | - | - | - | 0.5 | CORE_SW |
| FREQM | 0x4400C000 | 10: DONE<br>10: WINMON<br>- | CLKMSK0.10 (APB) | 0 | 2: MSR0<br>3: MSR1<br>4: REF | 10 | 4: DONE<br>5: WINMON<br>- | 0: START | R | | 0.6 | CORE_SW |
| WDT | 0x4400E000 | 11: EW_A | CLKMSK0.11 (APB) | 0 | - | 11 | - | - | - | - | 0.7 | CORE_SW |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| HMATRIX2 | 0x44010000 | 47: HMATRIX | CLKMSK0.30 (AHB) | 0 | - | 22 | - | - | - | - | 0.8 | CORE_SW |
| | | | CLKMSK0.31 (APB) | 0 | | | | | | | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| EIC | 0x44020000 | - | CLKMSK0.13 (APB) | 0 | 5: EIC | 13 | 21: EXTINT0 | - | - | - | 0.16 | CORE_SW |
| | | 16: EXTINT0 | | | | | 22: EXTINT1 | | | | | |
| | | 17: EXTINT1 | | | | | 23: EXTINT2 | | | | | |
| | | 18: EXTINT2 | | | | | 24: EXTINT3 | | | | | |
| | | 19: EXTINT3 | | | | | 25: EXTINT4 | | | | | |
| | | 20: EXTINT4 | | | | | 26: EXTINT5 | | | | | |
| | | 21: EXTINT5 | | | | | 27: EXTINT6 | | | | | |
| | | 22: EXTINT6 | | | | | 28: EXTINT7 | | | | | |
| | | 23: EXTINT7 | | | | | 29: EXTINT8 | | | | | |
| | | 24: EXTINT8 | | | | | 30: EXTINT9 | | | | | |
| | | 25: EXTINT9 | | | | | 31: EXTINT10 | | | | | |
| | | 26: EXTINT10 | | | | | 32: EXTINT11 | | | | | |
| | | 27: EXTINT11 | | | | | 33: EXTINT12 | | | | | |
| | | 28: EXTINT12 | | | | | 34: EXTINT13 | | | | | |
| | | 29: EXTINT13 | | | | | 35: EXTINT14 | | | | | |
| | | 30: EXTINT14 | | | | | 36: EXTINT15 | | | | | |
| | | 31: EXTINT15 | | | | | - | | | | | |
| | | 32: NSCHK | | | | | - | | | | | |
| PAC | 0x44022000 | - | CLKMSK0.14 (APB) | 0 | - | 14 | - | - | - | - | 0.17 | CORE_SW |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| MBISTINTF | 0x44028000 | - | CLKMSK0.18 (APB) | 0 | - | - | - | - | - | - | 0.20 | CORE_SW |
| | | | DCLK: 0.19 | 2 | | | | | | | | |
| MCRAMC | 0x4402C000 | 34: INT | - | - | - | 16 | - | - | - | - | 0.22 | CORE_SW |
| | | | - | - | | | | | | | | |
| CAN0 | 0x4402E000 | 35: INT0 | CLKMSK0.23 (AHB) | 0 | 6: CAN0 | 17 | - | - | - | - | 0.23 | CORE_SW |
| | | 36: INT1 | | | | | | | | | | |
| | | 37: BERR | | | | | | | | | | |
| | | 37: DBG | | | | | | | | | | |
| CAN1 | 0x44030000 | 38: INT0 | CLKMSK0.24 (AHB) | 0 | 7: CAN1 | 18 | - | - | - | - | 0.24 | CORE_SW |
| | | 39: INT1 | | | | | | | | | | |
| | | 40: BERR | | | | | | | | | | |
| | | 40: DBG | | | | | | | | | | |
| H2PB0 | 0x44032000 | - | CLKMSK0.25 (AHB) | 0 | - | 19 | - | - | - | - | 0.25 | CORE_SW |
| | | | CLKMSK0.26 (APB) | 0 | | | | | | | | |
| **H2PB1 Bridge Peripherals** | | | | | | | | | | | | |
| PORT | 0x44800000 | 41: NSCHK0 | CLKMSK0.27 (APB) | 0 | - | 20 | - | 2: EV0 | R | - | 1.0 | CORE_SW |
| | | 41: NSCHK1 | | | | | | 3: EV1 | R | | | |
| | | 41: NSCHK2 | | | | | | 4: EV2 | R | | | |
| | | 41: NSCHK3 | | | | | | 5: EV3 | R | | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| DMAC | 0x44802000 | 42: TCMPL0, SUSP0, TERR0 | CLKMSK0.28 (AHB) | 0 | - | 21 | 37: CH0 | 6: CH0 | R | - | 1.1 | CORE_SW |
| | | 43: TCMPL1, SUSP1, TERR1 | | | | | 38: CH1 | 7: CH1 | R | | | |
| | | 44: TCMPL2, SUSP2, TERR2 | | | | | 39: CH2 | 9: CH2 | - | | | |
| | | 45: TCMPL3, SUSP3, TERR3 | | | | | 40: CH3 | 9: CH3 | | | | |
| | | 46: TCMPL4, SUSP4 TERR4 | | | | | - | - | | | | |
| | | 46: TCMPL5, SUSP5, TERR5 | | | | | - | - | | | | |
| | | 46: TCMPL6, SUSP6, TERR6 | | | | | - | - | | | | |
| | | 46: TCMPL7, SUSP7, TERR7 | | | | | - | - | | | | |
| BROMC | 0x44804000 | - | CLKMSK1.0 (AHB) | 0 | - | 23 | - | - | - | - | 1.2 | CORE_SW |
| | | | CLKMSK1.1 (APB) | 0 | | | | | | | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| EVSYS | 0x44806000 | 48: EVD0 | CLKMSK1.3 (APB) | 0 | 8: CH0 | 25 | - | - | - | - | 1.3 | CORE_SW |
| | | 48: OVR0 | | | 9: CH1 | | | | | | | |
| | | 49: EVD1 | | | 10: CH2 | | | | | | | |
| | | 49: OVR1 | | | 11: CH3 | | | | | | | |
| | | 50: EVD2 | | | 12: CH4 | | | | | | | |
| | | 50: OVR2 | | | 13: CH5 | | | | | | | |
| | | 51: EVD3 | | | 14: CH6 | | | | | | | |
| | | 51: OVR3 | | | 15: CH7 | | | | | | | |
| | | 52: EVD4 | | | 16: CH8 | | | | | | | |
| | | 52: OVR4 | | | 17: CH9 | | | | | | | |
| | | 53: EVD5 | | | 18: CH10 | | | | | | | |
| | | 53: OVR5 | | | 19: CH11 | | | | | | | |
| | | 54: EVD6 | | | - | | | | | | | |
| | | 54: OVR6 | | | - | | | | | | | |
| | | 55: EVD7 | | | - | | | | | | | |
| | | 55: OVR7 | | | - | | | | | | | |
| | | 56: EVD8 | | | - | | | | | | | |
| | | 56: OVR8 | | | - | | | | | | | |
| | | 57: EVD9 | | | - | | | | | | | |
| | | 57: OVR9 | | | - | | | | | | | |
| | | 58: EVD10 | | | - | | | | | | | |
| | | 58: OVR10 | | | - | | | | | | | |
| | | 59: EVD11 | | | - | | | | | | | |
| | | 59: OVR11 | | | - | | | | | | | |
| SERCOM0 | 0x44808000 | 60: 6 | CLKMSK1.4 (APB) | 0 | 20: SLOW | 26 | - | - | - | 4: RX | 1.4 | CORE_SW |
| | | 61: 5 | | | 21: CORE | | | | | 5: TX | | |
| | | 62: 0 | | | - | | | | | - | | |
| | | 63: 1 | | | - | | | | | - | | |
| | | 64: 2 | | | - | | | | | - | | |
| | | 65: 3 | | | - | | | | | - | | |
| | | 66: 4 | | | - | | | | | - | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| SERCOM1 | 0x4480A000 | 67: 6 | CLKMSK1.5 (APB) | 0 | 20: SLOW | 27 | - | - | - | 6: RX | 1.5 | CORE_SW |
| | | 68: 5 | | | 22: CORE | | | | | 7: TX | | |
| | | 69: 0 | | | - | | | | | - | | |
| | | 70: 1 | | | - | | | | | - | | |
| | | 71: 2 | | | - | | | | | - | | |
| | | 72: 3 | | | - | | | | | - | | |
| | | 73: 4 | | | - | | | | | - | | |
| SERCOM2 | 0x4480C000 | 74: 6 | CLKMSK1.6 (APB) | 0 | 20: SLOW | 28 | - | - | - | 8: RX | 1.6 | CORE_SW |
| | | 75: 5 | | | 23: CORE | | | | | 9: TX | | |
| | | 76: 0 | | | - | | | | | - | | |
| | | 77: 1 | | | - | | | | | - | | |
| | | 78: 2 | | | - | | | | | - | | |
| | | 79: 3 | | | - | | | | | - | | |
| | | 80: 4 | | | - | | | | | - | | |
| SERCOM3 | 0x4480E000 | 81: 6 | CLKMSK1.7 (APB) | 0 | 20: SLOW | 29 | - | - | - | 10: RX | 1.7 | CORE_SW |
| | | 82: 5 | | | 24: CORE | | | | | 11: TX | | |
| | | 83: 0 | | | - | | | | | - | | |
| | | 84: 1 | | | - | | | | | - | | |
| | | 85: 2 | | | - | | | | | - | | |
| | | 86: 3 | | | - | | | | | - | | |
| | | 87: 4 | | | - | | | | | - | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| TCC0 | 0x44810000 | 88: DFS | CLKMSK1.8 (APB) | 0 | 25: TCC01 | 30 | 41: OVF | 10: EV0 | R | 12: OVF | 1.8 | CORE_SW |
| | | 88: ERR | | | | | 42: TRG | 11: EV1 | R | 13: MC0 | | |
| | | 88: FAULTA | | | | | 43: CNT | 12: MC0 | R | 14: MC1 | | |
| | | 88: FAULTB | | | | | 44: MC0 | 13: MC1 | R | - | | |
| | | 88: FAULT0 | | | | | 45: MC1 | - | - | - | | |
| | | 88: FAULT1 | | | | | - | - | - | - | | |
| | | 88: OVF | | | | | - | - | - | - | | |
| | | 88: UFS | | | | | - | - | - | - | | |
| | | 89: CNT | | | | | - | - | - | - | | |
| | | 89: TRG | | | | | - | - | - | - | | |
| | | 90: MC0 | | | | | - | - | - | - | | |
| | | 91: MC1 | | | | | - | - | - | - | | |
| TCC1 | 0x44812000 | 92: DFS | CLKMSK1.9 (APB) | 0 | 25: TCC01 | 31 | 46: OVF | 14: EV0 | R | 15: OVF | 1.9 | CORE_SW |
| | | 92: ERR | | | | | 47: TRG | 15: EV1 | R | 16: MC0 | | |
| | | 92: FAULTA | | | | | 48: CNT | 16: MC0 | R | 17: MC1 | | |
| | | 92: FAULTB | | | | | 49: MC0 | 17: MC1 | R | - | | |
| | | 92: FAULT0 | | | | | 50: MC1 | - | - | - | | |
| | | 92: FAULT1 | | | | | - | - | - | - | | |
| | | 92: OVF | | | | | - | - | - | - | | |
| | | 92: UFS | | | | | - | - | - | - | | |
| | | 93: CNT | | | | | - | - | - | - | | |
| | | 93: TRG | | | | | - | - | - | - | | |
| | | 94: MC0 | | | | | - | - | - | - | | |
| | | 95: MC1 | | | | | - | - | - | - | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| TCC2 | 0x44814000 | 96: DFS | CLKMSK1.10 (APB) | 0 | 26: TCC2 | 32 | 51: OVF | 18: EV0 | R | 18: OVF | 1.10 | CORE_SW |
| | | 96: ERR | | | | | 52: TRG | 19: EV1 | R | 19: MC0 | | |
| | | 96: FAULTA | | | | | 53: CNT | 20: MC0 | R | 20: MC1 | | |
| | | 96: FAULTB | | | | | 54: MC0 | 21: MC1 | R | - | | |
| | | 96: FAULT0 | | | | | 55: MC1 | - | - | - | | |
| | | 96: FAULT1 | | | | | - | - | - | - | | |
| | | 96: OVF | | | | | - | - | - | - | | |
| | | 96: UFS | | | | | - | - | - | - | | |
| | | 97: CNT | | | | | - | - | - | - | | |
| | | 97: TRG | | | | | - | - | - | - | | |
| | | 98: MC0 | | | | | - | - | - | - | | |
| | | 99: MC1 | | | | | - | - | - | - | | |
| TCC3 | 0x44816000 | 100: DFS | CLKMSK1.11 (APB) | 0 | 27: TCC3 | 33 | 56: OVF | 22: EV0 | R | 21: OVF | 1.11 | CORE_SW |
| | | 100: ERR | | | | | 57: TRG | 23: EV1 | R | 22: MC0 | | |
| | | 100: FAULTA | | | | | 58: CNT | 24: MC0 | R | 23: MC1 | | |
| | | 100: FAULTB | | | | | 59: MC0 | 25: MC1 | R | - | | |
| | | 100: FAULT0 | | | | | 60: MC1 | - | - | - | | |
| | | 100: FAULT1 | | | | | - | - | - | - | | |
| | | 100: OVF | | | | | - | - | - | - | | |
| | | 100: UFS | | | | | - | - | - | - | | |
| | | 101: CNT | | | | | - | - | - | - | | |
| | | 101: TRG | | | | | - | - | - | - | | |
| | | 102: MC0 | | | | | - | - | - | - | | |
| | | 103: MC1 | | | | | - | - | - | - | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| ADC | 0x44818000 | 130: REQ0 | CLKMSK1.17 (APB) | 0 | 33: ADC | 39 | 76: RESRDY | 38: TRIG0 | A | 37: PFFRDY | 1.12 | CORE_SW |
| | | 131: REQ1 | | | | | 77: CMP | 39: TRIG1 | A | - | | |
| | | - | | | | | - | 40: TRIG2 | A | - | | |
| | | - | | | | | - | 41: TRIG3 | A | - | | |
| | | - | | | | | - | 42: TRIG4 | A | - | | |
| | | - | | | | | - | 43: TRIG5 | A | - | | |
| | | - | | | | | - | 44: TRIG6 | A | - | | |
| | | - | | | | | - | 45: TRIG7 | A | - | | |
| | | - | | | | | - | 46: TRIG8 | A | - | | |
| | | - | | | | | - | 47: TRIG9 | A | - | | |
| | | - | | | | | - | 48: TRIG10 | A | - | | |
| SERCOM4 | 0x44820000 | 104: 6 | CLKMSK1.12 (APB) | 0 | 28: CORE | 34 | - | - | - | 24: RX | 1.16 | CORE_SW |
| | | 105: 5 | | | | | | | | 25: TX | | |
| | | 106: 0 | | | | | | | | - | | |
| | | 107: 1 | | | | | | | | - | | |
| | | 108: 2 | | | | | | | | - | | |
| | | 109: 3 | | | | | | | | - | | |
| | | 110: 4 | | | | | | | | - | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| SERCOM5 | 0x44822000 | 111: 6 | CLKMSK1.13 (APB) | 0 | 29: CORE | 35 | - | - | - | 26: RX | 1.17 | CORE_SW |
| | | 112: 5 | | | | | | | | 27: TX | | |
| | | 113: 0 | | | | | | | | - | | |
| | | 114: 1 | | | | | | | | - | | |
| | | 115: 2 | | | | | | | | - | | |
| | | 116: 3 | | | | | | | | - | | |
| | | 117: 4 | | | | | | | | - | | |
| TCC4 | 0x44824000 | 118: DFS | CLKMSK1.14 (APB) | 0 | 30: TCC4 | 36 | 61: OVF | 26: EV0 | R | 28: OVF | 1.18 | CORE_SW |
| | | 118: ERR | | | | | 62: TRG | 27: EV1 | R | 29: MC0 | | |
| | | 118: FAULTA | | | | | 63: CNT | 28: MC0 | R | 30: MC1 | | |
| | | 118: FAULTB | | | | | 64: MC0 | 29: MC1 | R | - | | |
| | | 118: FAULT0 | | | | | 65: MC1 | - | - | - | | |
| | | 118: FAULT1 | | | | | - | - | - | - | | |
| | | 118: OVF | | | | | - | - | - | - | | |
| | | 118: UFS | | | | | - | - | - | - | | |
| | | 119: CNT | | | | | - | - | - | - | | |
| | | 119: TRG | | | | | - | - | - | - | | |
| | | 120: MC0 | | | | | - | - | - | - | | |
| | | 121: MC1 | | | | | - | - | - | - | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| TCC5 | 0x44826000 | 122: DFS | CLKMSK1.15 (APB) | 0 | 31: TCC5 | 37 | 66: OVF | 30: EV0 | R | 31: OVF | 1.19 | CORE_SW |
| | | 122: ERR | | | | | 67: TRG | 31: EV1 | R | 32: MC0 | | |
| | | 122: FAULTA | | | | | 68: CNT | 32: MC0 | R | 33: MC1 | | |
| | | 122: FAULTB | | | | | 69: MC0 | 33: MC1 | R | - | | |
| | | 122: FAULT0 | | | | | 70: MC1 | - | - | - | | |
| | | 122: FAULT1 | | | | | - | - | - | - | | |
| | | 122: OVF | | | | | - | - | - | - | | |
| | | 122: UFS | | | | | - | - | - | - | | |
| | | 123: CNT | | | | | - | - | - | - | | |
| | | 123: TRG | | | | | - | - | - | - | | |
| | | 124: MC0 | | | | | - | - | - | - | | |
| | | 125: MC1 | | | | | - | - | - | - | | |
| TCC6 | 0x44828000 | 126: DFS | CLKMSK1.16 (APB) | 0 | 32: TCC6 | 38 | 71: OVF | 34: EV0 | R | 34: OVF | 1.20 | CORE_SW |
| | | 126: ERR | | | | | 72: TRG | 35: EV1 | R | 35: MC0 | | |
| | | 126: FAULTA | | | | | 73: CNT | 36: MC0 | R | 36: MC1 | | |
| | | 126: FAULTB | | | | | 74: MC0 | 37: MC1 | R | - | | |
| | | 126: FAULT0 | | | | | 75: MC1 | - | - | - | | |
| | | 126: FAULT1 | | | | | - | - | - | - | | |
| | | 126: OVF | | | | | - | - | - | - | | |
| | | 126: UFS | | | | | - | - | - | - | | |
| | | 127: CNT | | | | | - | - | - | - | | |
| | | 127: TRG | | | | | - | - | - | - | | |
| | | 128: MC0 | | | | | - | - | - | - | | |
| | | 129: MC1 | | | | | - | - | - | - | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| AC | 0x4482A000 | 132: COMP0 | CLKMSK1.18 (APB) | 0 | 34: AC | 40 | 78: COMP0 | 49: SOC0 | R | - | 1.21 | CORE_SW |
| | | 132: COMP1 | | | | | 79: COMP1 | 50: SOC1 | R | | | |
| | | 132: WIN0 | | | | | 80: WIN0 | - | - | | | |
| | | - | | | | | - | - | - | | | |
| PTC | 0x4482C000 | 133: EOC | CLKMSK1.19 (APB) | 0 | 35: PTC | 41 | 81: EOC | 51: DSEQR | A | 38: EOC | 1.22 | CORE_SW |
| | | 133: WCOMP | | | | | 82: WCOMP | 52: STCONV | A | 39: SEQ | | |
| | | 133: ACRRDY | | | | | - | - | - | 40: WCOMP | | |
| | | - | | | | | - | - | - | - | | |
| CCL0 | 0x4482E000 | | CLKMSK1.20 (APB) | 0 | 36: CCL0 | 42 | 83: LUTOUT0 | 53: LUTIN0 | R | - | 1.23 | CORE_SW |
| | | | | | | | 84: LUTOUT1 | 54: LUTIN1 | R | | | |
| | | | | | | | 85: LUTOUT2 | 55: LUTIN2 | R | | | |
| | | | | | | | 86: LUTOUT3 | 56: LUTIN3 | R | | | |
| CCL1 | 0x44830000 | | CLKMSK1.21 (APB) | 0 | 37: CCL1 | 43 | 87: LUTOUT0 | 57: LUTIN0 | R | - | 1.24 | CORE_SW |
| | | | | | | | 88: LUTOUT1 | 58: LUTIN1 | R | | | |
| | | | | | | | 89: LUTOUT2 | 59: LUTIN2 | R | | | |
| | | | | | | | 90: LUTOUT3 | 60: LUTIN3 | R | | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| USB | 0x44832000 | 134: EORSMDNRSM | CLKMSK1.22 (AHB) | 00 | 38: USB | 44 | - | - | - | - | 1.25 | CORE_SW |
| | | 134: EORSTRST | CLKMSK1.23 (APB) | | | | | | | | | |
| | | 134: LPMDCONN | - | | | | | | | | | |
| | | 134: LPMSUSPDDISC | - | | | | | | | | | |
| | | 134: MSOF | - | | | | | | | | | |
| | | 134: RAMACER | - | | | | | | | | | |
| | | 134: RXSTPTXSTP7..0 | - | | | | | | | | | |
| | | 134: STALL0STALL7..0 | - | | | | | | | | | |
| | | 134: STALL17..10 | - | | | | | | | | | |
| | | 134: SUSPEND | - | | | | | | | | | |
| | | 134: TRFAIL0TRFAIL7..0 | - | | | | | | | | | |
| | | 134: TRFAIL1PERR7..0 | - | | | | | | | | | |
| | | 134: UPRSM | - | | | | | | | | | |
| | | 134: WAKEUP | - | | | | | | | | | |
| | | 135: SOFHSOF | - | | | | | | | | | |
| | | 136: TRCPT07..00 | - | | | | | | | | | |
| | | 137: TRCPT17..10 | - | | | | | | | | | |
| H2PB1 | 0x44838000 | - | CLKMSK1.25 (AHB) | 0 | - | 46 | - | - | - | - | 1.28 | CORE_SW |
| | | | CLKMSK1.26 (APB) | | | | | | | | | |


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

**Table 9-2.** Peripherals Configuration Summary (continued)

| Peripheral name | Base address | NVIC IRQ Index | MCLK | | GCLK Peripheral Channel Index (PCHCTRLx) | PAC Peripheral Identifier Index (WRCTRL. PERID) | Events (EVSYS) | | | DMA Trigger Source Index (CHCTRLB. TRIGSRC) | H2PBx. NONSECy index | Power Domain |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | (CLKMSKx.y) | CLKDIVx | | | Generators (CHANNELn. EVGEN) | Users (USERm) | Path Type | | | |
| **H2PB2 Bridge Peripherals** | | | | | | | | | | | | |
| PM | 0x45000000 | 2: SLEEPRDY | CLKMSK0.3 (APB) | 1 | - | 3 | - | - | - | - | 2.0 | CORE_SW |
| SUPC | 0x45002000 | 3: LVDDET<br>3: LVDRDY<br>3: ULDORDY | CLKMSK0.4 (APB) | 1 | | 4 | 1: LVDET | | | | 2.1 | CORE_SW |
| RSTC | 0x45004000 | - | CLKMSK0.5 (APB) | 1 | - | 5 | - | - | - | - | 2.2 | CORE_SW |
| RTC | 0x45006000 | 12: TAMPER<br>13: OVF<br>14: PER7..0<br>15: CMP3..0<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>-<br>- | CLKMSK0.12 (APB) | 1 | | 12 | 6: PER0<br>7: PER1<br>8: PER2<br>9: PER3<br>10: PER4<br>11: PER5<br>12: PER6<br>13: PER7<br>14: CMP0<br>15: CMP1<br>16: CMP2<br>17: CMP3<br>18: TAMPER<br>19: OVF<br>20: PERD | 1: TAMPER | A | 3: TIMESTAMP | 2.3 | CORE_BU |
| OSC32KCTRL | 0x45008000 | 7: XOSC32KRDY<br>8: XOSC32KFAIL | CLKMSK0.7 (APB) | 1 | - | 7 | 3: XOSC32KFAIL | - | - | - | 2.4 | CORE_BU |
| H2PB2 | 0x4500E000 | - | CLKMSK1.27 (AHB) | 0 | - | 47 | - | - | - | - | 2.7 | CORE_SW |
| | | | CLKMSK1.28 (APB) | 1 | | | | | | | | |


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

# 10. Processor and Architecture

## 10.1. Processor and Architecture

Refer to the Arm reference documents *"ARM Cortex-M23 Processor Data Sheet"* and *"ARM Cortex-M23 Technical Reference Manual"*, available for download at the following location: www.arm.com.

## 10.2. Arm Cortex-M23 Configuration

**Table 10-1.** Arm Cortex-M23 Configuration

| Feature | Configuration |
|---|---|
| ARMv8-M security (Trust Zone) | Yes |
| Memory Protection (MPU) | Regions: 8 Non-Secure and 8 Secure |
| Number of interrupt priority levels | 2 |
| Number of breakpoints comparators | 4 |
| Number of watch point comparators | 2 |

**Note:** TrustZone® is not available on GC devices.

## 10.3. Nested Vector Interrupt Controller

The Nested Vectored Interrupt Controller (NVIC) provides the user interface to control the external and on-chip peripheral interrupts. The NVIC provides up to 240 interrupt sources (unused sources are reserved for future use) that can be programmed with up to 4 or 8 different priority levels. The NVIC is included as part of the CPU. For additional information, refer to the "Arm Core Technical Reference Manual" available for download at www.arm.com.

## 10.4. NVIC Interrupt Mapping

**Table 10-2.** NVIC Interrupt Mapping

| Module | Interrupt Index : Source |
|---|---|
| EIC – External Interrupt Controller | NMI : NMI |
| FCR – Flash Read Controller | 0 : FCR |
| FCW – Flash Write Controller | 1 : FCW |
| PM – Power Manager | 2 : SLEEPRDY |
| SUPC – Supply Controller | 3 : LVDDET/LVDRDY/ULDORDY |
| OSCCTRL – Oscillators Control | 4 : XOSCRDY/XOSCFAIL/CLKFAIL<br>5 : DFLLRDY/DFLLLOCK/DFLLOVF/DFLLUNF/DFLLRCS/DFLLFAIL<br>6 : PLLLOCKR_0/PLLLOCKF_0 |
| OSC32KCTRL – 32 kHz Oscillators Control | 7 : XOSC32KRDY<br>8 : XOSC32KFAIL |
| MCLK – Main Clock Controller | 9 : CKRDY |
| FREQM – Frequency Meter | 10 : DONE/WINMON |
| WDT – Watchdog Timer | 11 : EW_A |
| RTC – Real-Time Counter | 12 : TAMPER<br>13 : OVF<br>14 : PER0/PER1/PER2/PER3/PER4/PER5/PER6/PER7<br>15 : CMP0/CMP1/CMP2/CMP3 |


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

**Table 10-2.** NVIC Interrupt Mapping (continued)

| Module | Interrupt Index : Source |
|---|---|
| EIC – External Interrupt Controller | 16 : EXTINT0<br>17 : EXTINT1<br>18 : EXTINT2<br>19 : EXTINT3<br>20 : EXTINT4<br>21 : EXTINT5<br>22 : EXTINT6<br>23 : EXTINT7<br>24 : EXTINT8<br>25 : EXTINT9<br>26 : EXTINT10<br>27 : EXTINT11<br>28 : EXTINT12<br>29 : EXTINT13<br>30 : EXTINT14<br>31 : EXTINT15<br>32 : NSCHK |
| MCRAMC – Multi-Channel RAM Controller | 34 : INT |
| CAN0 – Control Area Network | 35 : INT0<br>36 : INT1<br>37 : BERR/DBG |
| CAN1 – Control Area Network | 38 : INT0<br>39 : INT1<br>40 : BERR/DBG |
| PORT – Port Module | 41 : NSCHK0/NSCHK1/NSCHK2/NSCHK3 |
| DMAC – Direct Memory Access Controller | 42 : TCMPL0/SUSP0/TERR0<br>43 : TCMPL1/SUSP1/TERR1<br>44 : TCMPL2/SUSP2/TERR2<br>45 : TCMPL3/SUSP3/TERR3<br>46 : TCMPL4/SUSP4/TERR4/TCMPL5/SUSP5/TERR5/TCMPL6/SUSP6/TERR6/TCMPL7/SUSP7/TERR7 |
| HMATRIX2 – HSB Matrix | 47 : HMATRIX |
| EVSYS – Event System Interface | 48 : EVD0/OVR0<br>49 : EVD1/OVR1<br>50 : EVD2/OVR2<br>51 : EVD3/OVR3<br>52 : EVD4/OVR4<br>53 : EVD5/OVR5<br>54 : EVD6/OVR6<br>55 : EVD7/OVR7<br>56 : EVD8/OVR8<br>57 : EVD9/OVR9<br>58 : EVD10/OVR10<br>59 : EVD11/OVR11 |


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

**Table 10-2.** NVIC Interrupt Mapping (continued)

| Module | Interrupt Index : Source |
|---|---|
| SERCOM0 – Serial Communication Interface | 60 : ERROR<br>61 : RXBRK<br>62 : DRE/PREC/MB<br>63 : TXC/AMATCH/SB<br>64 : RXC/DRDY<br>65 : RXS/SSL/TXFE<br>66 : CTSIC/RXFF |
| SERCOM1 – Serial Communication Interface | 67 : ERROR<br>68 : RXBRK<br>69 : DRE/PREC/MB<br>70 : TXC/AMATCH/SB<br>71 : RXC/DRDY<br>72 : RXS/SSL/TXFE<br>73 : CTSIC/RXFF |
| SERCOM2 – Serial Communication Interface | 74 : ERROR<br>75 : RXBRK<br>76 : DRE/PREC/MB<br>77 : TXC/AMATCH/SB<br>78 : RXC/DRDY<br>79 : RXS/SSL/TXFE<br>80 : CTSIC/RXFF |
| SERCOM3 – Serial Communication Interface | 81 : ERROR<br>82 : RXBRK<br>83 : DRE/PREC/MB<br>84 : TXC/AMATCH/SB<br>85 : RXC/DRDY<br>86 : RXS/SSL/TXFE<br>87 : CTSIC/RXFF |
| TCC0 – Timer Counter for Control Applications | 88 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS<br>89 : CNT/TRG<br>90 : MC0<br>91 : MC1 |
| TCC1 – Timer Counter for Control Applications | 92 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS<br>93 : CNT/TRG<br>94 : MC0<br>95 : MC1 |
| TCC2 – Timer Counter for Control Applications | 96 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS<br>97 : CNT/TRG<br>98 : MC0<br>99 : MC1 |
| TCC3 – Timer Counter for Control Applications | 100 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS<br>101 : CNT/TRG<br>102 : MC0<br>103 : MC1 |


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

**Table 10-2.** NVIC Interrupt Mapping (continued)

| Module | Interrupt Index : Source |
|---|---|
| SERCOM4 – Serial Communication Interface | 104 : ERROR<br>105 : RXBRK<br>106 : DRE/PREC/MB<br>107 : TXC/AMATCH/SB<br>108 : RXC/DRDY<br>109 : RXS/SSL/TXFE<br>110 : CTSIC/RXFF |
| SERCOM5 – Serial Communication Interface | 111 : ERROR<br>112 : RXBRK<br>113 : DRE/PREC/MB<br>114 : TXC/AMATCH/SB<br>115 : RXC/DRDY<br>116 : RXS/SSL/TXFE<br>117 : CTSIC/RXFF |
| TCC4 – Timer Counter for Control Applications | 118 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS<br>119 : CNT/TRG<br>120 : MC0<br>121 : MC1 |
| TCC5 – Timer Counter for Control Applications | 122 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS<br>123 : CNT/TRG<br>124 : MC0<br>125 : MC1 |
| TCC6 – Timer Counter for Control Applications | 126 : DFS/ERR/FAULTA/FAULTB/FAULT0/FAULT1/OVF/UFS<br>127 : CNT/TRG<br>128 : MC0<br>129 : MC1 |
| ADC – ADC Controller | 130 : REQ0<br>131 : REQ1 |
| AC – Analog Comparator Controller | 132 : COMP0/COMP1/WIN0 |
| PTC – Peripheral Touch Controller | 133 : EOC/WCOMP/ACRRDY |
| USB – Full-Speed Universal Serial Bus | 134 : EORSMDNRSM/EORSTRST/LPMDCONN/LPMSUSPDDISC/MSOF/RAMACER/ RXSTPTXSTP0/RXSTPTXSTP1/<br>RXSTPTXSTP2/RXSTPTXSTP3/RXSTPTXSTP4/RXSTPTXSTP5/RXSTPTXSTP6/RXSTPTXSTP7/ STALL0STALL0/<br>STALL0STALL1/STALL0STALL2/STALL0STALL3/STALL0STALL4/STALL0STALL5/STALL0STALL6/ STALL0STALL7/<br>STALL10/STALL11/STALL12/STALL13/STALL14/STALL15/STALL16/STALL17/SUSPEND/ TRFAIL0TRFAIL0/<br>TRFAIL0TRFAIL1/TRFAIL0TRFAIL2/TRFAIL0TRFAIL3/TRFAIL0TRFAIL4/TRFAIL0TRFAIL5/ TRFAIL0TRFAIL6/<br>TRFAIL0TRFAIL7/TRFAIL1PERR0/TRFAIL1PERR1/TRFAIL1PERR2/TRFAIL1PERR3/ TRFAIL1PERR4/TRFAIL1PERR5/<br>TRFAIL1PERR6/TRFAIL1PERR7/UPRSM/WAKEUP<br>135 : SOFHSOF<br>136 : TRCPT00/TRCPT01/TRCPT02/TRCPT03/TRCPT04/TRCPT05/TRCPT06/TRCPT07<br>137 : TRCPT10/TRCPT11/TRCPT12/TRCPT13/TRCPT14/TRCPT15/TRCPT16/TRCPT17 |


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

## 10.5. High Speed Bus

The Bus Matrix implements a multi-layer AHB (Advanced High-performance Bus), based on the AHB-Lite protocol, that enables parallel access paths between multiple AHB hosts and clients in a system, therefore increasing the overall bandwidth. The Bus Matrix interconnects up to 16 AHB hosts to up to 16 AHB clients. The normal latency to connect a host to a client is one cycle except for the default host of the accessed client which is connected directly (zero cycle latency).

## 10.6. Bus Matrix Connectivity

**Table 10-3.** Connectivity Between Initiators and Targets

| | | Host # | Bus Clients | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | FCR | MCRAMC | | | BROMC | APB0 | APB1 | APB2 |
| | | | | AHB0 | AHB1 | AHB2 | | | | |
| Bus Hosts | Client | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | CORE | 0 | x | x | | | x | x | x | x |
| | DMAC | 1 | x | | | x | | x | x | x |
| | CAN0 | 2 | | | x | | | | | |
| | CAN1 | 3 | | | x | | | | | |
| | FCW | 4 | | | x | | | | | |

**Note:**

1. Connectivity between a host and client is represented by the cells with an “x” in this table. If the cell is empty, there is no connectivity between the host and client. If there is connectivity the cell will contain a “x”.


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

# 11. Memories

## 11.1. Embedded Memories

The following are key features of the embedded memories:

- Flash memories are made of 4 KB pages
- 512 KB in-system self-programmable Flash (PFM) with Error Correction Code (ECC)
- 16 KB of Boot Flash Memory (BFM)
- 32 KB of Configuration Flash Memory (CFM)
  - 1 User OTP page (USEROTP)
  - 1 ROM Configuration page (ROMCFG)
  - 1 Calibration OTP page (CALOTP)
  - 2 Boot Configuration pages (BOOTCFG1/1A)
- 128 KB SRAM with ECC

## 11.2. Physical Memory Map

The high-speed bus is implemented as a bus matrix. All high-speed bus addresses are fixed and are never remapped in any way, even during boot. The 32-bit physical address space is mapped as described in the Product Mapping chapter.

## 11.3. SRAM Memory Configuration

### Retention

In Standby and Hibernate modes, all SRAM is retained.

### RAM Error Correction

For safety applications, the PIC32CM GC00 family of devices embed SRAM error correction codes (ECC) to detect and correct single bit errors, or to enable dual-error detection for the system memory. The ECC feature can be enabled or disabled after reset through the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse.

## 11.4. Flash Memories

This devices comes with a 512 KB Program Flash Memory (PFM), a 16 KB Boot Flash Memory (BFM), and a 32 KB Configuration Flash memory (CFM). They all support ECC for correcting on the fly ECC errors or detecting double errors.

### 11.4.1. Program and Boot Flash Memory (PFM and BFM)

Refer to the "Non-Volatile Memory Controller" Chapter for details about Programming, Erasing, Reading these memories.

### 11.4.2. Configuration Flash Memory (CFM)

The CFM contains the following pages:

- **CALOTP:** This page contains read only calibration values. Some values need to be copied by the application software in ADC, PTC, or USB before these peripherals can be used.
- **ROMCFG:** This page contains:
  - The Debug Access Level (DAL) value
  - The Boot ROM interactive mode command keys and locks
  - Pre-boot configuration settings for running MBIST on Boot or Boot ROM CRC


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

- **BOOTCFG1/1A:** These pages contain the boot configuration selected by the user and applied by the ROM code. They must contain the same duplicate information. The ROM code will automatically switch from one to the other in case of errors found.
- **USEROTP:** 1 page dedicated for user application. It could be used to store a serial number, some user defined keys, or custom analog calibration values, and so on. Once written, each row can be locked independently forever, refer to the FCW.UOWP for additional information. Once protected and locked, it cannot be erased using a page erase or chip erase.

### 11.4.3. Boot Configuration

The BOOTCFG1 and BOOTCFG1A Boot configuration pages contain the following:

- Configuration settings
- Security settings
- Firmware metadata

The BOOTCFG1 and BOOTCFG1A share the same mapping and must be programmed identically. The redundant copy allows the copy to switch to the second page if an error is found in the first one.

The BOOTCFG page mapping is detailed in the following table:

**Table 11-1.** BOOTCFG Page Mapping

| Page | Address Range | Description |
|---|---|---|
| BOOTCFG1A | 0x0A000000 | Firmware Meta Data |
| | 0x0A0007FF | |
| | 0x0A000800 | Registers |
| | 0x0A0008D0 | |
| BOOTCFG1 | 0x0A002000 | Firmware Meta Data |
| | 0x0A0027FF | |
| | 0x0A002800 | Registers |
| | 0x0A0028D0 | |

### 11.4.4. ROMCFG Configuration

The DAL fuses in the ROMCFG page allows configuring the Debugger Access Level. Refer to the DSU for details.

All ROMCFG fuses are described in the following sections.

### 11.4.5. CAL-OTP Configuration

The following values are programmed at the factory and need to be copied by the application software in the associated peripheral registers.

**Table 11-2.** Factory Programmed Values to be Copied in Peripheral Registers

| Address | Name | Description |
|---|---|---|
| 0x0A007180 | FCCFG64 | Application software must copy FCCFG64[7:0] in PTC.CALIB |
| 0x0A007184 | FCCFG65 | Application software must copy FCCFG65[31:0] in ADC.ADCCTRL |
| 0x0A007190 | FCCFG68 | Application software must copy FCCFG68[15:0] in USB.PADCAL |

## 11.5. Unique ID (UID)

The PIC32CM GC00 provides a 128 bit (UNIQID*n*, *n* = 0,1,2,3) which conforms to the version 4 UUID format. See the following table for the location in Flash memory. The unique ID is programmed onto the device at manufacture.

For more information on the standard, refer to the RFC4122 at tools.ietf.org/html/rfc4122.


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

**Table 11-3.** Unique ID Map

| Address | Name | Description |
|---|---|---|
| 0x0A0071E0 | UNIQID0 | UNIQID[31:0] |
| 0x0A0071E4 | UNIQID1 | UNIQID[63:32] |
| 0x0A0071E8 | UNIQID2 | UNIQID[95:64] |
| 0x0A0071EC | UNIQID3 | UNIQID[127:96] |


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

# 12. Boot ROM

## 12.1. Boot ROM

The Boot ROM is the entry point of the device following any type of reset. It is responsible for initializing the system and ensuring the integrity of the configuration of the device before transferring control to the first mutable application code.

The Boot ROM performs a series of vital checks, including image validation and configuration verification. If these checks pass, the CPU loads the initial Program Counter (PC) and Stack Pointer (SP) values from Flash memory and begins executing user code.

**Note:** Before transferring control to the user application, the Boot ROM may reset the SRAM region from 0x2000_0000 to 0x2000_AFFF. Users should avoid relying on any pre-initialized data in this region while developing the first mutable application code. SRAM region 0x2001_0000 to 0x2001_FFFF remains untouched by the Boot ROM and can be used for persistent data storage across resets, if required.

**Important:** Throughout this document, wherever **Debugger** is mentioned, it shall be interpreted as SWD-Host and vice-versa.

## 12.2. Features

The Boot ROM in the PIC32CM GC00 device includes several built-in features to support a reliable booting.

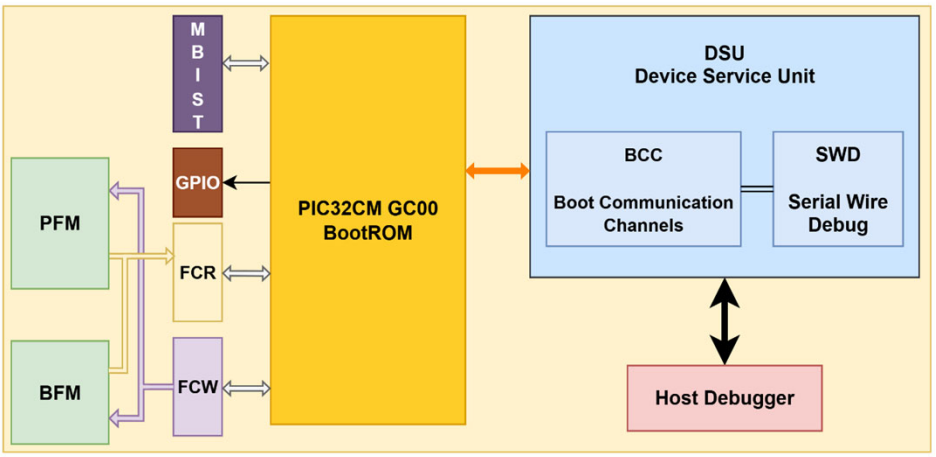
- **Device Integrity Check:** Verifies the authenticity and integrity of the boot image.
- **Boot Source Selection:** Determines the boot source based on fuse settings and configuration.
- **Device Lockdown:** Supports complete device lockdown, preventing any external read/write access.
- **Boot Failure Indication:** Asserts a configurable user I/O pin if boot fails when fault signaling is enabled via the BOOTCFG1/1A fuse (BOOT_GPIOSEL).
- **Boot Interactive Mode:** Allows access to maintenance operations (e.g., CFM integrity check, chip erase) when a debugger probe is connected.
- **Debug Access and Security:**To maintain a secure boot process, the Boot ROM disables all debug access during its execution.
- **Clock Configuration:**During Boot ROM execution, the device clocks remain unchanged and operate at their default values. The Boot ROM does not modify the clock configuration.

## 12.3. Block Diagram

The following figure illustrates a conceptual block diagram of the PIC32CM GC00 Boot ROM. This diagram represents the typical functional structure of the Boot ROM and is intended for illustrative purposes only. It does not reflect the internal implementation details of the Boot ROM IP.


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

**Figure 12-1.** PIC32 GC00 Boot ROM Block Diagram




## 12.4. Boot ROM Dependencies

The Boot ROM is designed to operate based on user-configured boot settings defined in the BOOTCFG1/1A and ROMCFG fuses. No external hardware connections are required for Boot ROM operation.

**Note:** Users must ensure that the BOOT_GPIOSEL fuse is configured in a way that does not interfere with the overall hardware functionality of the application.

### 12.4.1. Clocks

The Boot ROM executes using the device's default clock source: a 48 MHz DFLL (Digital Frequency Locked Loop) operating in open-loop mode. This clock is automatically enabled by the hardware following any type of reset, such as Power-on Reset, External Reset, and Watchdog Reset.

This 48 MHz DFLL is the fixed and non-configurable clock source for Boot ROM execution. It ensures consistent and deterministic behavior during the early boot phase, prior to any user-configurable clock initialization.

The Boot ROM does not support switching to alternative clock sources. Any clock configuration changes must be performed by the user application after control is transferred from the Boot ROM.

### 12.4.2. Device Fuse Programming (BOOTCFG 1, BOOTCFG1A & ROMCFG)

The Boot ROM reads the BOOTCFG1/1A and ROMCFG fuses and applies the selected user configuration to the device.

**⚠ WARNING** Incorrect fuse settings may prevent the device from booting successfully. In extreme cases, invalid fuse configurations may render the device permanently non-functional.

### 12.4.3. Debug Operations

For security reasons, debug access is disabled during Boot ROM execution, except when the device enters Park Mode. The Boot ROM supports both cold-plugging and hot-plugging, allowing a debugger to attempt entry into Boot Interactive Mode under specific conditions. Refer to the *DSU* chapter for additional information on cold and hot plugging mechanisms.


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

## 12.5. Functional Description

### 12.5.1. PIC32CM Boot ROM

The Boot ROM is the entry point of the device following any type of reset. It is responsible for performing essential device checks and applying user-defined configurations before transferring control to the first mutable application code. The Boot ROM operation can be logically divided into the following functional stages.

#### 12.5.1.1. Device Fuse Integrity Check

Validates the integrity of critical fuse areas: USERCFG1/1A, ROMCFG, and CALOTP. The Boot ROM performs integrity checks on the USERCFG1/1A, ROMCFG, and CALOTP fuse regions. If any integrity violation is detected, the Boot ROM halts further execution and places the device into a known safe state called Boot Interactive Mode. This mode is described in detail in a later section.

Once fuse integrity is confirmed, the Boot ROM proceeds to apply the user-defined configuration settings.

#### 12.5.1.2. Debugger Probe Detection

The Boot ROM is designed to asynchronously accept a cold-plugging request from a debugger (i.e. SWD host) at any time. The Boot ROM automatically senses debugger attachment following a successful cold-plug.

#### 12.5.1.3. Application of User Configuration

Applies user-selected fuse configurations after successful integrity check. Details on available user configuration can be found from BootCFG1/1A and ROMCFG fuse details.

#### 12.5.1.4. First Mutable Application Firmware Integrity Check (if enabled)

Performs an optional integrity check on the user application code, based on fuse settings and firmware meta data programmed by the user.

#### 12.5.1.5. Fault Signaling (if enabled)

Asserts a user-configurable fault signal if an error is detected during the boot process, provided the user has enabled fault signaling via BOOTCFG 1/1A fuse called BOOT_GPIOSEL.

#### 12.5.1.6. Transition to First Mutable Application Executable (FME)

Transfers control to the first mutable application executable which has to reside at 0x0800_0000 address of BFM by loading the PC and SP values only if boot operation is successful.

### 12.5.2. Debugger Access Level (DAL)

The PIC32CM Boot ROM supports two levels of debugger access, known as Debugger Access Levels (DALs). These levels define the degree of access a debugger has to the device during and after boot:

- DAL2: Full access. The device is completely open, allowing all operations supported by the device. This level is typically used during development and debugging.
- DAL0: Locked state. The device is highly restricted. Debug access is limited and defined by the user prior to setting DAL to 0.

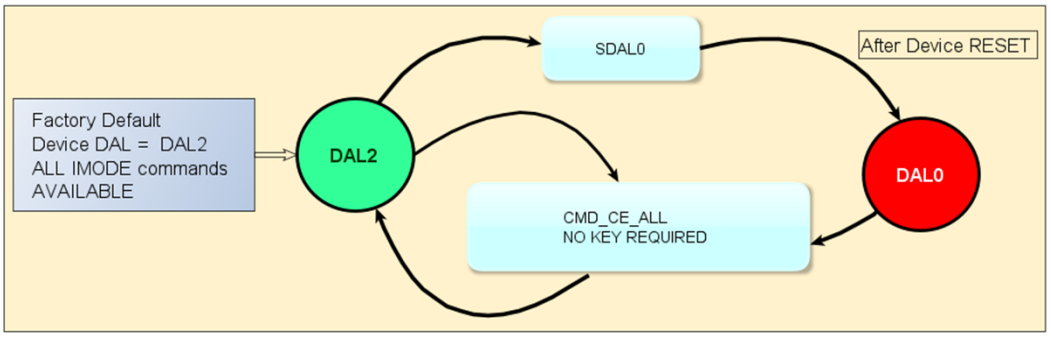
The active DAL is determined by the value programmed into the ROMCFG DAL fuse. Once set, the DAL governs the debugger's ability to interact with the device, including access to memory, peripherals, and Boot ROM features. Further details on DAL behavior, transitions, and impact on device access are provided below.

#### 12.5.2.1. Factory Default Value of DAL

The PIC32CM devices are factory programmed with DAL2. Once the device is programmed with DAL0, the DAL level can be raised by issuing CMD_CE_ALL when device Boot ROM is in interactive mode. Alternatively CMD_HOST_DALELEV command could be used to elevate the device DAL if CMD_HOST_DALELEV is not locked.


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

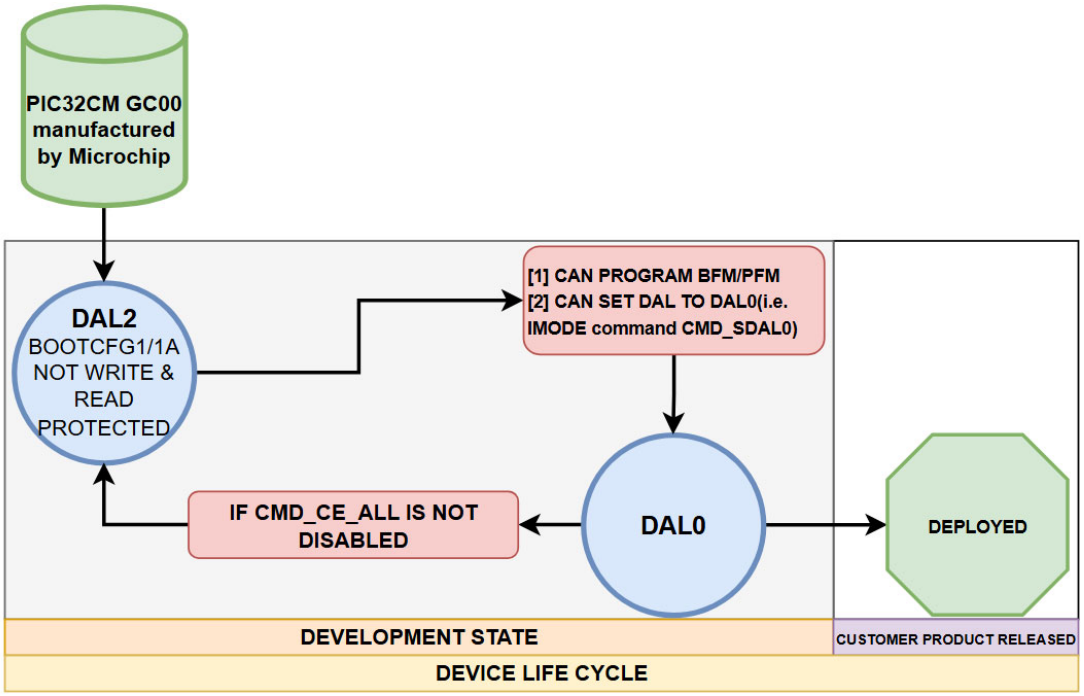
**Figure 12-2.** Factory Default Value of DAL




### 12.5.2.2. Device DAL Life Cycle

The Device Life Cycle has primarily two states: Development and Deployed. The Development state provides debug and programming access to the device. Once application development completes, the device can be placed into the Deployed state by programming the Debug Access Level to DAL0. If the Boot ROM commands for CE_ALL and CMD_HOST_DALELEV (see Table 12-2) are not locked, and USERCFG1/1A and ROMCFG fuse memory regions are not write protected, the device can move back to the Development state. If these commands are disabled, and the memory regions locked, then the Development state is unrecoverable from the debug interface.

**Figure 12-3.** DAL Life Cycle




**CAUTION** When the device is set to DAL0 and if the CMD_CE_ALL (Chip Erase All) command is locked, the device may become permanently unrecoverable if the user does not retain access to the required KEYVAL_ fuse needed to unlock CMD_CE_ALL.


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

**Note:** In the Development state, the ROMCFG page which holds the Debugger Access Level (DAL) can be modified. It is recommended that when the device is programmed for the production phase, that the user selects the write protect (and write lock option) for the ROMCFG page (defined in USERCFG1/1A fuse). Doing so prevents internal firmware from erasing ROMCFG and programming the device to DAL2 (Development).

### 12.5.3. PIC32CM Boot ROM Operating Modes

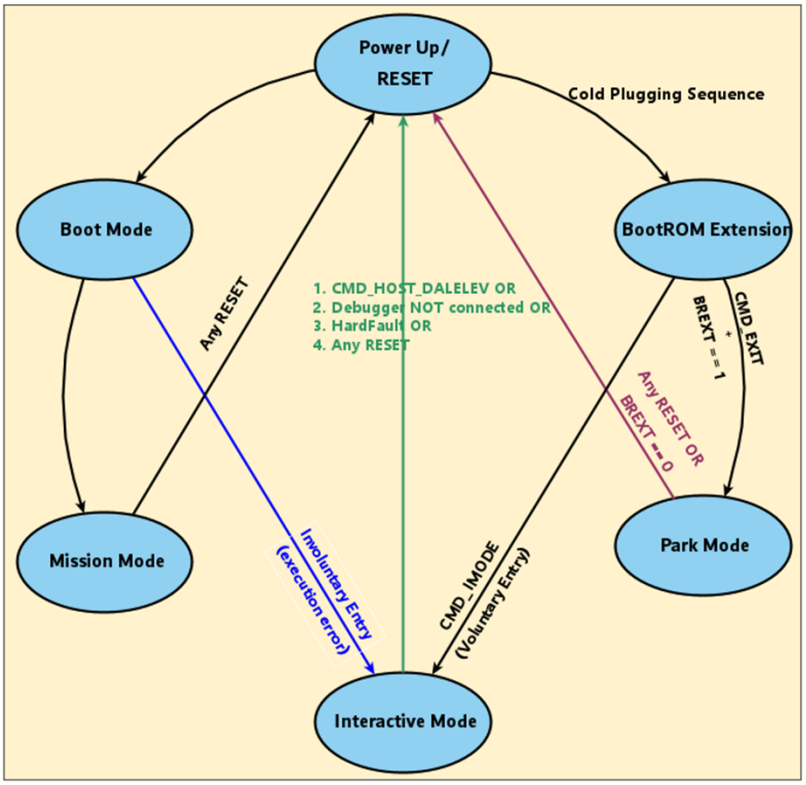
The Boot ROM operation is logically divided into the following four modes, each serving a distinct purpose during the device startup and debug process:

1. **Boot Mode:**
   This is the default mode entered immediately after any type of reset. In this mode, the Boot ROM performs system initialization, fuse integrity checks, and attempts to boot the device using the user-configured settings.
2. **Interactive Mode (IMODE):**
   The Boot ROM enters Interactive Mode under the following conditions:
   - A boot failure occurs (Example: due to fuse integrity check failure or invalid configuration)
   - A debugger explicitly requests entry into IMODE to perform advanced operations

   In this mode, users can perform tasks such as:
   - Full chip erase
   - Device locking (setting DAL = 0)
   - Integrity checks on CFM, PFM, ROM, SRAM
3. **Mission Mode:**
   This mode allows the device to begin execution from the first mutable application executable stored at 0x0800_0000 in the Boot Flash Memory region (BFM). The device remains in mission mode until the device is reset due to any reason. Once the device is reset, the device goes into Boot Mode.


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

**Figure 12-4.** Diagram of Modes



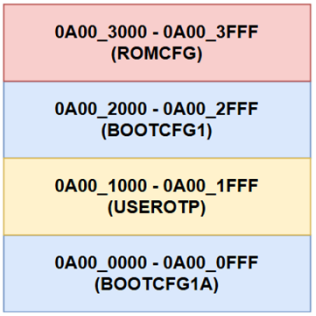

**Note:** Interactive Mode is specifically designed to enable the host debugger to offer additional debug features.

### 12.5.4. PIC32CM GC00 Configuration Flash Memory Layout

The PIC32CM GC00 Boot ROM utilizes a predefined Configuration Fuse Memory (CFM) layout to manage boot-time settings. This layout includes dual copies of the boot configuration, providing redundancy to enhance reliability.

The Boot ROM attempts to boot the device as long as at least one of the BOOTCFG memory regions is valid and passes integrity checks. However, it is strongly recommended that both BOOTCFG fuse regions be programmed to take full advantage of the available redundancy and ensure robust boot behavior.

**Figure 12-5.** Configuration Flash Memory Layout





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

#### 12.5.4.1. Device USERCFG 1/1A and ROMCFG Fuses

The PIC32CM GC00 Boot ROM supports complete device configuration through the User Configuration Fuses (USERCFG1/1A). To enhance reliability of the device, the Boot ROM maintains two physical copies of the user configuration:

- USERCFG1
- USERCFG1A

The Boot ROM selects one of these fuse sets during the boot process, selected by configuring the SEQ and SEQBAR fuses. If both USERCFG1 and USERCFG1A fail integrity checks, the Boot ROM halts the boot process and places the device into interactive mode.

**Selection of USERCFG1 v/s USERCFG1A**

The Boot ROM uses Firmware Metadata (FWMD) provided by the user to determine which boot configuration to use. This metadata includes two 32-bit unsigned integers:

- SEQ0: Associated with USERCFG1
- SEQ1: Associated with USERCFG1A

These sequence values are used to determine which configuration is more recent or preferred.

**Important:** Microchip provides firmware metadata generator tool. Do not attempt to program SEQ0/SEQ1 without using Microchip provided firmware metadata generator tool. Users need to use the FWMD tool by selecting the target device and generate the required FWMD for the target device. The FWMD tool is available for download from www.microchip.com.

The Boot ROM selects the boot configuration based on the following conditions:

**Table 12-1.** Boot Rom Conditions

| Condition | Selected USERCFG fuse for Boot |
|---|---|
| SEQ0 > SEQ1 | USERCFG1 |
| SEQ1 > SEQ0 | USERCFG1A |
| SEQ0 = SEQ1 and both ≠ 0xFFFFFFFF | Boot Process halted, Boot ROM goes to Interactive mode |
| SEQ0 = SEQ1 = 0xFFFFFFFF | Boots using default USERCFG values |

**Note:** SEQ0 = SEQ1 must not be programmed to the same value.

#### 12.5.4.2. Device ROMCFG Fuses

The ROMCFG region contains critical fuses that allow users to configure device access, enable self-tests, and enforce security policies. These fuses are evaluated by the Boot ROM during the early boot phase. Below is a summary of the key fuses within the ROMCFG region.

##### 12.5.4.2.1. DAL Fuse

This fuse determines the Debugger Access Level (DAL) that becomes effective after the Boot ROM completes its execution and transfers control to the user application.

- **DAL2**: Full access (value = 0xFFFFFF55)
- **DAL0**: Locked state (any value other than 0xFFFFFF55)

#### 12.5.4.3. FRCFGBROM Fuse

This fuse enables an **internal self-test** of the Boot ROM memory contents. If enabled and the self-test fails, the device halts further execution.

- **Default State**: Disabled after any reset


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

- **User Action**: Must be explicitly enabled via fuse programming

#### 12.5.4.4. FRCFGMBIST Fuse

This fuse enables Memory Built-In Self-Test (MBIST) on the entire device SRAM. If MBIST is enabled (e.g., MOBB = 0x1 or 0x2), the Boot ROM checks the MBIST result during boot.

- If MBIST Fails: The Boot ROM halts the boot process and enters an infinite loop followed by a software reset.
- If MBIST Passes: Boot continues normally.

**Note:** MBIST is an intrusive operation for on-chip SRAM and hence, content of SRAM will not be persistent after the device is reset.

#### 12.5.4.5. KEYVAL_ Fuses

The ROMCFG region includes multiple KEYVAL_ fuses used for device-specific configuration and security enforcement.

### 12.5.5. Boot Failure Signal

The PIC32CM GC00 Boot ROM provides a mechanism to indicate boot failure through a user-configurable Boot Failure Signal, enabled through the BOOT_GPIOSEL fuse located in the BOOTCFG1/1A region.

Users can select a specific I/O pin to serve as the fault indicator for their application. If a boot failure occurs (for example, CRC check failure on FME), the Boot ROM asserts the selected I/O pin to signal the fault condition.

**Note:** By default, the Boot Failure Signal is disabled (BOOT_GPIOSEL = NONE). To enable this feature, users must program the BOOT_GPIOSEL fuse with a valid I/O selection. Refer to the *Fault Signal Configuration* for additional information on available I/O options and configuration guidelines.

**Important:** Once the Boot ROM transfers control to the first mutable application code, it no longer manages or controls the fault signal pin. The application firmware is responsible for any further handling or de-assertion of the signal.

### 12.5.6. Error Handling and Fault Conditions

Upon any reset, the Boot ROM begins execution by performing a fuse integrity check, which is a critical prerequisite before the device can proceed to execute the first mutable application code. During the boot process, the device may encounter unsafe operating conditions due to various reasons, including but not limited to:

- Invalid or inconsistent device configuration
- Memory faults (double-bit ECC errors)
- Unintended or undefined device states

In such scenarios, the Boot ROM proactively halts the boot process and transitions the device into a known safe state called Interactive Mode (Example: involuntary entry to IMODE). In this mode, user intervention is required to perform corrective actions.

**Note:** Upon any reset, the device begins execution with Error Correction Code (ECC) enabled for both Flash and SRAM. Users may choose to disable ECC checking for specific memory regions via the BOOTCFG1/1A configuration fuses. However, this user-defined ECC configuration becomes effective only after the Boot ROM completes execution and transfers control to the first mutable application code at address 0x0800_0000. During Boot ROM execution, ECC checks remain always enabled, regardless of user configuration.


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

### 12.5.7. Device Fuse Integrity Check

As described in *PIC32CM GC00 Boot ROM Operating Modes*, the Boot ROM performs fuse integrity checks automatically upon any type of reset. This behavior is always enabled and does not require any user configuration.

If the user has programmed the sequence numbers (SEQ0 and SEQ1) using the Microchip-provided Firmware Metadata (FWMD) Generator Tool, the Boot ROM uses these values to determine the preferred boot configuration between BOOTCFG1 and BOOTCFG1A.

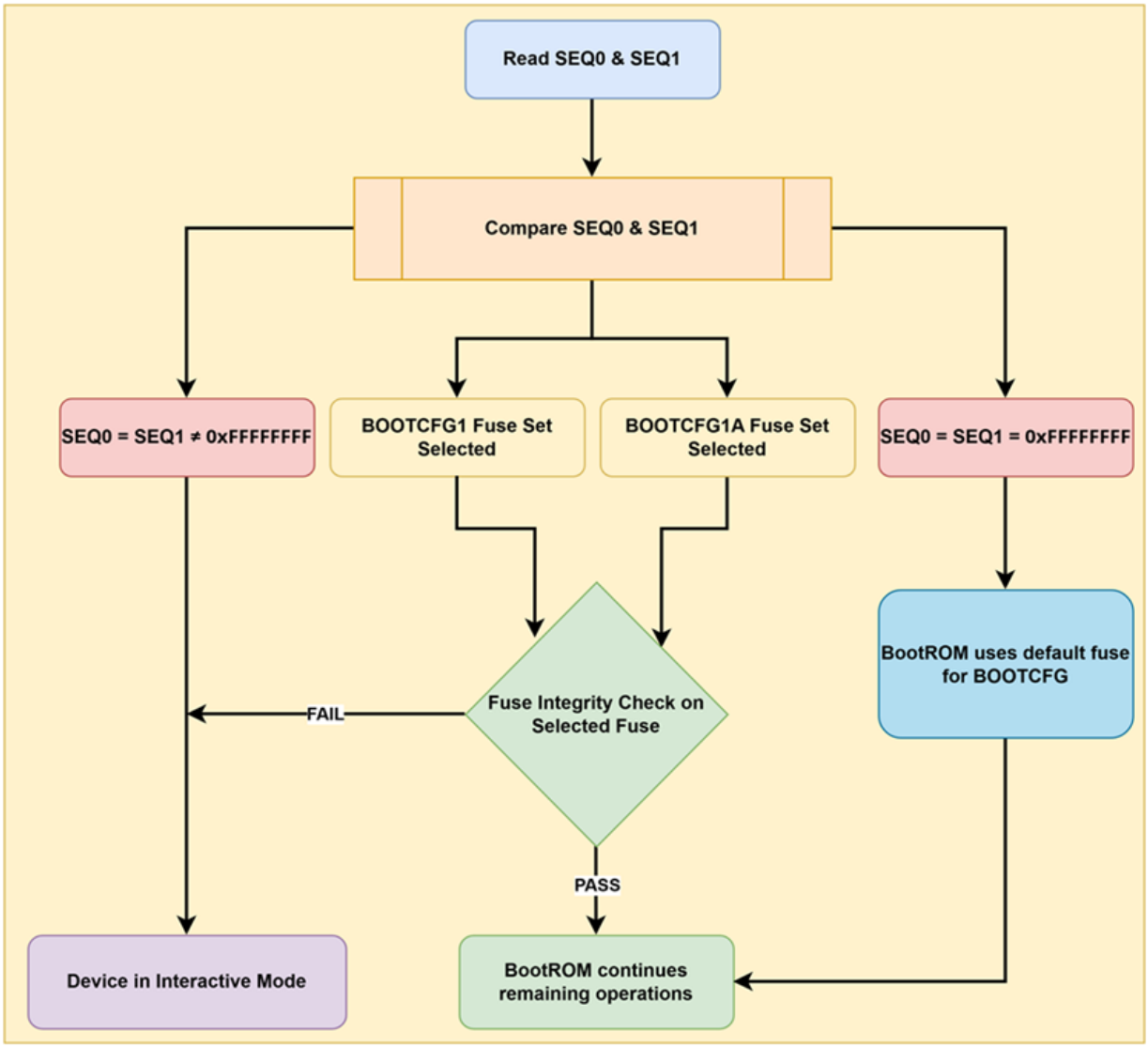
The fuse integrity check process follows this sequence:

1. The Boot ROM reads SEQ0 and SEQ1 to determine the user preferred boot configuration.
2. It performs an integrity check on the selected BOOTCFG fuse set.
3. If the selected fuse set fails the integrity check, the Boot ROM automatically attempts to boot using the redundant BOOTCFG fuse set.
4. If both fuse sets fail integrity checks, the Boot ROM halts the boot process and places the device into a known safe state called interactive mode,

**Note:** If the boot process fails due to fuse integrity check failure, the Boot Failure Signal (configured through BOOT_GPIOSEL) is not asserted.


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

**Figure 12-6.** USERCFG1/1x fuse Integrity Check Selection




## 12.5.8. Peripheral Configuration and Initialization

### 12.5.8.1. FCW and FCR

The user can optionally choose to lock and write protect ROMCFG and USERCFG1/1A fuse memory as outlined in *Configuration Flash Memory Layout*. Refer to the USERCFG1/1A configuration fuse details provided in *Device USERCFG 1/1A Fuse*.

The PIC32CM device supports Error Correction Code (ECC) for the user non-volatile memory. The supported ECC algorithm is Single Error Correction (SEC) and Double Error Detection (DED). The user can optionally choose to enable/disable ECC feature. Refer to the FUSES_FCR_ ECCCTRL fuse in USERCFG1/1A for additional information on ECC feature enable/disable option. The ECC check is by default enabled during the device boot operation and user selected ECC feature is applied by the Boot ROM before executing the first mutable executable application code.

### 12.5.8.2. WDT

The user can optionally choose to configure WDT as required by the target first mutable executable code. This feature is useful in cases where the first mutable executable code needs WDT peripheral


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

as soon as the device begins execution of FME (first mutable executable) application code. The user should ensure WDT time out is programmed with a value such that WDT timeout is at least 50 mSec.

### 12.5.9. Usage of Boot ROM Interactive Mode

As mentioned earlier in this chapter, the Boot ROM entry to Interactive Mode (IMODE) may be voluntarily or involuntarily. Involuntary entry to IMODE signals that the Boot ROM experienced error and/or fault during the boot operation. On the other hand, voluntary entry to IMODE means users have connected a debugger to the device and request for IMODE.

Involuntary entry to IMODE – In this operating scenario, the user is required to connect a debugger to the device in order to make use of the supported IMODE commands.

Voluntary entry to IMODE – In this operating scenario, the user connected a debugger device and the device is ready to accept one of the supported Interactive Mode commands. Voluntary entry to IMODE is typically done during product development or debugging post-product released related issues.

Once the device is in IMODE, the following Interactive Mode commands are supported for PIC32CM GC00 device. The user needs to use one of the supported programmer/debuggers to communicate with PIC32CM GC00 Boot ROM over Interactive Mode.

A Cold Plugging sequence is mandatory to enter the Interactive Mode. The PIC32CM GC00 device is designed to accept cold plugging requests from the SWD Host at any time during its operation.

The following debuggers supports Interactive Modes:

- Microchip designed debugger, such as PKOB, PICkit4, ICE4, ICD5 which are available at https://www.microchip.com
- Segger https://www.segger.com
- IAR https://www.iar.com
- Keil https://www.keil.com

**Note:** For additional information, contact a local Microchip sales team or contact www.microchip.com/support to obtain design information about Boot ROM Interactive Modes communication.

**Table 12-2.** IMODE Commands

| Sr. No | IMODE Command Name | IMODE Command Purpose |
|---|---|---|
| **1** | CMD_IMODE | Debugger requests the device to enter Interactive Mode (IMODE) |
| **2** | CMD_EXIT | Debugger requests the device to reset and begin executing the first mutable executable code or device enters into Park Mode when the DSU_STATUSA.BREXT0 bit is set |
| **3** | CMD_SDAL0 | Sets the Device Access Level (DAL) to 0(locked state) |
| **4** | CMD_HOST_DALELEV | Elevates the Device Access Level to DAL 2(access as per BOOTCFG1/1A and ROMCFG fuse values) |
| **5** | CMD_CE_ALL | Performs a full device erase, restoring factory default. Refer to table 12-3 to know more about factory default state. |
| **6** | CMD_READCFM | Reads Configuration Fuse Memory (CFM) contents |
| **7** | CMD_CRC | Performs CRC check on a specified memory region |


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

**Note:**
IMODE commands **6** and **7** are considered advanced debugging features of the PIC32CM GC00 device.

These commands are typically not required during standard product development and are not intended for post-production use. As such, support for these commands is optional in host debugger implementations.

To use these commands, the debugger must explicitly support the corresponding IMODE command protocol.

Once the user has connected one of the supported debuggers to PIC32CM GC00, the user can use one of the following interactive mode commands mentioned in the following sections.

#### 12.5.9.1. CMD_IMODE and CMD_EXIT

The Boot ROM supports two key commands to manage interactions between device and external debugger. These commands are typically issued by the SWD host debugger and are not directly invoked by the end use.

**CMD_IMODE**

- After performing a Cold-Plugging sequence, this command is issued by the debugger to request the device to enter Interactive Mode.
- No manual intervention is required from the user.
- The debugger handles all necessary communication with the PIC32CM GC00 device to initiate this mode.
- IMODE is typically used for advanced debugging, fuse programming, or device inspection operations.

**CMD_EXIT**

- Upon receiving CMD_EXIT, the Boot ROM resumes the normal boot flow and transfers control to the first mutable application executable code.
- The application execution begins at BFM address 0x0800_0000.

#### 12.5.9.2. CMD_SDAL0 – Set Device Access Level to DAL0

The CMD_SDAL0 command allows the user to permanently set the device's Device Access Level (DAL) to DAL0.

Once the device is in DAL0, the only ways to restore it to DAL2 are as follows:

- Issuing a chip erase using CMD_CE_ALL (provided this command is not locked) or
- Using the CMD_HOST_DALELEV command if the CMD_HOST_DALELEV is not locked
- Write ROMCFG DAL from user code to revert to DAL2

Users must ensure they have access to the necessary recovery mechanisms before executing CMD_SDAL0, especially in production environments.

#### 12.5.9.3. CMD_HOST_DALELEV - Elevates the Device Access Level to DAL 2

The CMD_HOST_DALELEV command allows the debugger to temporarily elevate the device's Device Access Level (DAL) to DAL2, provided this command has not been locked by the user.

**Behavior and Scope**

- When executed, the device transitions to DAL2 access level.
- This elevated access level is temporary and remains valid only until the device undergoes a Brown-out Reset (BOR) or Power-on Reset (POR).
- The elevated DAL2 level becomes effective after a reset (excluding BOR or POR) and is applied only when the device begins executing the first mutable application code.


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

- While the device remains in Interactive Mode, the DAL remains at DAL0, regardless of any command execution

This command is useful for temporarily unlocking access to protected features during development or pre-production, but it does not persist across full power cycles or brown-out events.

#### 12.5.9.4. CMD_CE_ALL - Performs a Full Device Erase

The CMD_CE_ALL command performs a full chip erase, restoring the PIC32CM GC00 device to its factory default state. This operation erases all major memory regions and resets configuration fuses to their default values.

**Table 12-3.** Memory Types

| Sr. No | Memory Type | ERASED? |
|---|---|---|
| **1** | BFM | YES |
| **2** | PFM | YES |
| **3** | BOOTCFG1(1) | YES |
| **4** | BOOTCFG1A(1) | YES |
| **5** | ROMCFG(2) | YES |
| **6** | SRAM | YES |

**Notes:**

1. Firmware Metadata (FWMD) is stored within BOOTCFG1/1A and is erased as part of this operation.
2. After erase operation, the DAL fuse is reset to its default value: DAL2.
3. Erased contents of memory is 0xFFFFFFFF except for SRAM erased value is 0x00000000.
4. If the user programs an application in FME without programming USERCFGx in BOOTCFGx, the BootROM uses a default set of values. Refer to the section 12.7 to know more about the default values.

**⚠ CAUTION** Users must exercise extreme caution when locking the CMD_CE_ALL (Chip Erase All) IMODE command. It is strongly recommended that users apply prudent attention before disabling or restricting access to this command through fuse settings. Locking CMD_CE_ALL may permanently prevent the device erasure, which can lead to irreversible loss of reprogramming capability, especially in conjunction with restrictive DAL and KEYVAL_CE_ALLx fuse configuration.

### 12.5.10. Unrecoverable Device Conditions

The PIC32CM device incorporates advanced security and intellectual property protection mechanisms. While these features are essential for safeguarding deployed applications, improper configuration may render the device permanently non-programmable and non-recoverable.

This section outlines specific operating conditions that users must avoid unless the intention is to permanently lock the device for field deployment.

#### 12.5.10.1. Device Access Level and IMODE Command CMD_CE_ALL

Users must exercise extreme caution when programming the following fuses:

**Table 12-4.** Fuses that Require Caution

| Fuse/Command | Value to avoid device brick condition |
|---|---|
| ROMCFG_FUSES → DAL | DAL0 |
| IMODE command CMD_CE_ALL | LOCKED |


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

| Table 12-4. Fuses that Require Caution (continued) | |
|---|---|
| **Fuse/Command** | **Value to avoid device brick condition** |
| IMODE command CMD_HOST_DALELEV | LOCKED |

Programming the above configuration will permanently lock the device, disabling any future erase or reprogramming operations. This action should only be performed if the final goal is to prevent any further modification of the device once deployed in the field.

**⚠ CAUTION** Once the device is locked using the above fuse settings, failure analysis capabilities at Microchip are also disabled. In other words, if the device fails in the field, Microchip may be unable to provide support or diagnostics. By applying this configuration, the user acknowledges and accepts the risk of limited or no support from Microchip in case of device failure.

### 12.5.10.2. ROMCFG and BOOTCFG1/1A Read and Write Protection

Certain fuse configurations may render the device partially usable by restricting access to critical configuration fuses. Specifically, if the following fuses are programmed, the user will no longer be able to reprogram the ROMCFG and USERCFG1/1A fuses.

**Table 12-5.** ROMCFG and BOOTCFG1/1A Read and Write Protection

| ROMCFG/USERCFG Fuses | Value to avoid device brick condition |
|---|---|
| ROMCFG_FUSES → DAL | DAL2 |
| KEY_VAL of CMD_CE_ALL | Locked (0x00000000) |
| USERCFG_FUSES → FCW_CWP.RCWP | SET (= 1) |
| USERCFG_FUSES → FCW_CWP.RCWPLCK | SET (=1) |
| USERCFG_FUSES → FCW_CWP.BC1WP | SET (=1) |
| USERCFG_FUSES → FCW_CWP.BC1WPLCK | SET (=1) |

Once these write protections are enabled and locked, the device enters a restricted state where further modification of these fuses is not possible through standard methods.

### 12.5.10.3. MBIST Failure

The MBIST (Memory Built-In Self-Test) feature is a valuable safety mechanism offered by the PIC32CM GC00 device, particularly for applications requiring enhanced reliability of volatile user memory (SRAM). Once enabled through the ROMCFG_FUSES → FRCFGMBIST fuse, MBIST is executed automatically by the device during boot, without requiring any software or user intervention.

**MBIST Operation:**

- MBIST can be configured to run:
  - On every reset, or
  - Only after Power-on Reset (POR)
- Refer to the ROMCFG_FUSES section for details on selecting the MBIST-on-boot behavior

**Device Behavior on MBIST Failure:**

- If MBIST fails during boot, the Boot ROM halts further boot operations, rendering the device inaccessible
- The device does not proceed to execute any user application code, and no recovery is performed automatically


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

**Note:** In scenarios where the device does not boot, users can attempt to diagnose the issue by reading the DSU_REGS → BCC[1] register. To access this register, the user must connect a supported host debugger to the device through the SWD interface. It is essential that the host debugger supports capturing the DSU_REGS → BCC[1] register in such scenarios. Refer to the respective debugger's user guide/manual for detailed instructions on accessing and interpreting MBIST-related or other boot error conditions.

### 12.5.11. Recoverable Device Conditions

As mentioned in previous section where the user may configure PIC32CM GC00 device such that the device becomes unrecoverable, there are other scenarios where the device might become temporarily unrecoverable and with the help of the supported debugger, the user may be able to recover the device and bring the device back to normal usable condition.

When the DAL of the device is 2 and applying any of the following configurations will cause the device to temporarily become unavailable. To recover such a device, the steps outlined in the following table must be followed.

| Device DAL = 2 | | | |
|---|---|---|---|
| **CMD_CE_ALL** | **ROMCFG Write Protection (as selected through BOOTCFG Fuses)** | **BOOTCFG Write Protection (as selected through BOOTCFG Fuses)** | **Chip Erase Recovery Method** |
| **DISABLED** | FCW_CWP.RCWP == 0 AND<br>FCW_CWP.RCWPLCK == X | Don't Care | • Using Park Mode, do Page Erase ROMCFG fuse<br>• Reset the device<br>• Perform Chip Erase using IMODE<br>• Now the device is returned to normal state |
| **DISABLED** | FCW_CWP.RCWP == 1 AND<br>FCW_CWP.RCWPLCK == 1 | FCW_CWP.BC1/1A/1AWP == 0<br>AND<br>FCW_CWP.BC1/1ALCK == 0/1 | • Using Park Mode, Page Erase BOOTCFG using FCW<br>• Reset the device<br>• Page Erase ROMCFG.<br>• Perform IMODE Chip Erase |
| **DISABLED** | FCW_CWP.RCWP == 1<br>AND<br>FCW_CWP.RCWPLCK == 1 | FCW_CWP.BC1/1AWP == 1<br>AND<br>FCW_CWP.BC1/1ALCK == 0 | • Using Park Mode, ensure FCW_CWP.BC1/1AWP = 0<br>• Page Erase BOOTCFG<br>• Reset the device<br>• Page Erase ROMCFG<br>• Perform IMODE Chip Erase |

### 12.5.12. STATUS and ERROR Codes

During the boot process or while interacting with a host debugger, the device may emit a 32-bit code indicating either a STATUS or an ERROR condition. These codes are primarily intended to assist the host debugger in interpreting the device's state, especially during interactive mode.

**Note:** Park Mode is a specialized recovery mechanism and may be supported by the debugger tool. Refer to the debugger's user guide for instructions on entering Park Mode and performing recovery operations.

#### 12.5.12.1. STATUS Codes

STATUS codes reflect the current state or progress of the boot process. Some of these codes are internal and used by the debugger to support IMODE commands. However, one key STATUS code is relevant to user application is shown below:


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

| STATUS Code Name | Definition | Code |
|---|---|---|
| **STATUS_BOOTOK** | Indicates that the entire boot sequence has completed successfully. The Device Access Level (DAL) is set to the final value as determined by the ROMCFG fuse. | 0x00000004 |

Once the boot process completes successfully, the device writes the STATUS_BOOTOK code to the DSU_REGS->DSU_BCC[1] register. Optionally, this register can be read by the first mutable application executable code to confirm that the device has booted correctly.

**Note:** If the boot operation fails, the device will not transfer control to the first mutable executable (FME) application code. In such cases, the STATUS_BOOTOK code will not be written, and the device may remain in interactive mode.

### 12.5.12.2. ERROR Codes

During the boot process, the device may encounter error conditions that prevent it from completing initialization. In such cases, the Boot ROM issues a 32-bit error code, which is written to the DSU_REGS->DSU_BCC[1] register.

The DSU_BCC[1] register retains the last value written by the device, allowing the user or host debugger to inspect the cause of the boot failure.

The presence of an error code indicates that the boot operation was interrupted and did not proceed to the first mutable application executable code.

**Note:** The list of error codes provided below is not exhaustive. Only those error codes are specified which users might experience during the product life cycle. The probable causes listed for each error are indicative and may vary depending on system conditions.

#### 12.5.12.2.1. STATUS_ERR_HARDFAULT

CODE 0xEEEE0001

When the device encounters a HardFault during the boot operation, the Boot ROM issues the STATUS_ERR_HARDFAULT error code. This condition indicates a critical failure that prevents the device from completing the boot sequence.

**Probable Causes:**

While the exact cause may vary, one common scenario includes:

- Unrecoverable Memory Read Error
- Device malfunction due to unknown reason(s)

**Device Behavior:**

In response to this fault condition:

- The device enters a wait state for approximately 350 milliseconds
- After the wait period, the Boot ROM initiates a self-reset to attempt recovery
- This cycle of issue error code → wait → reset → retry continues indefinitely until the fault condition is resolved

This behavior ensures that transient faults, such as power instability or temporary memory corruption glitch do not permanently stall the device. However, persistent faults will result in repeated resets, which may be observable during system monitoring.

#### 12.5.12.2.2. STATUS_ERR_USERCFG

CODE 0xEEEE0004

This error indicates a failure in the boot configuration due to issues with the BOOTCFG1 and/or BOOTCFG1A fuses, or with the Firmware Metadata (FWMD) sequence numbers.

**Probable Causes**


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

- Invalid or corrupted boot configuration fuses
- The USERCFG1 or USERCFG1A fuses may contain memory faults or invalid values

**Device Behavior**

- Upon detecting this condition, the Boot ROM halts the boot process and issues the STATUS_ERR_USERCFG error code.
- The user may attempt to recover the device by:
  - Reprogramming the FWMD sequence numbers correctly
  - Ensuring valid values are written to USERCFG1 and USERCFG1A fuses
- If the error persists despite corrective actions, it indicates that the device is no longer usable due to permanent configuration faults

**STATUS_ERR_ROMCFG**
CODE 0xEEEE0019

The device has detected unrecoverable faults in the ROMCFG fuse, which is critical for determining boot behavior and security configuration

**Probable Causes**

- Double Memory Fault in ROMCFG fuse

**Device Behavior**

- Upon encountering this fault, the Boot ROM issues the STATUS_ERR_ROMCFG error code and halts the boot process
- The user may attempt to recover the device by reprogramming the ROMCFG fuse with valid values
- If the error persists after reprogramming, it indicates a permanent fault in the ROMCFG fuse, rendering the device no longer usable

#### 12.5.12.2.3. STATUS_MBIST_FAILED

CODE 0xEEEE0021

This error occurs when the MBIST on boot feature is enabled through the FRCFGMBIST fuse in ROMCFG and the device fails the MBIST (Memory Built-In Self-Test) during the boot process.

**Probable Causes:**

- MBIST failure on device SRAM. The device detected a fault in one or more SRAM regions during the MBIST execution.

**Device Behavior:**

- If MBIST fails, the device halts the boot process indefinitely and does not proceed to execute user application code.
- The user may attempt to recover the device by performing a Power-on Reset (POR) and observing whether the device resumes normal boot operation.
- This error condition is considered critical and may indicate a permanent fault in the memory subsystem.

**Note:** This error code `STATUS_MBIST_FAILED` is a special case which might require intervention from Microchip. If the issue persists, the user is advised to contact Microchip support for further diagnosis and resolution.


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

### 12.5.13. Key Values

The following table provides a list of keys that can be used to enable or disable specific Boot ROM command functionality. By default, the value of the keys at reset is 0xFFFF_FFFF. If the key is set to 0x0000_0000 the associated command will be disabled.

**Table 12-6.** Key Values

| Address | Key |
|---|---|
| 0x0A00_3420 | KEYVAL_IN- TCHK0 |
| 0x0A00_3424 | KEYVAL_IN- TCHK1 |
| 0x0A00_3428 | KEYVAL_IN- TCHK2 |
| 0x0A00_342C | KEYVAL_IN- TCHK3 |
| 0x0A00_3430 | KEYVAL_IN- TCHK4 |
| 0x0A00_3434 | KEYVAL_IN- TCHK5 |
| 0x0A00_3438 | KEYVAL_IN- TCHK6 |
| 0x0A00_343C | KEYVAL_IN- TCHK7 |
| 0x0A00_3440 | KEY- VAL_CE_ALL0 |
| 0x0A00_3444 | KEY- VAL_CE_ALL1 |
| 0x0A00_3448 | KEY- VAL_CE_ALL2 |
| 0x0A00_344C | KEY- VAL_CE_ALL3 |
| 0x0A00_3450 | KEY- VAL_CE_ALL4 |
| 0x0A00_3454 | KEY- VAL_CE_ALL5 |
| 0x0A00_3458 | KEY- VAL_CE_ALL6 |
| 0x0A00_345C | KEY- VAL_CE_ALL7 |
| 0x0A00_3500 | KEYVAL_S- DAL1_0 |
| 0x0A00_3504 | KEYVAL_S- DAL1_1 |
| 0x0A00_3508 | KEYVAL_S- DAL1_2 |
| 0x0A00_350C | KEYVAL_S- DAL1_3 |
| 0x0A00_3510 | KEYVAL_S- DAL1_4 |
| 0x0A00_3514 | KEYVAL_S- DAL1_5 |
| 0x0A00_3518 | KEYVAL_S- DAL1_6 |
| 0x0A00_351C | KEYVAL_S- DAL1_7 |
| 0x0A00_3520 | KEYVAL_S- DAL0_0 |
| 0x0A00_3524 | KEYVAL_S- DAL0_1 |
| 0x0A00_3528 | KEYVAL_S- DAL0_2 |
| 0x0A00_352C | KEYVAL_S- DAL0_3 |
| 0x0A00_3530 | KEYVAL_S- DAL0_4 |
| 0x0A00_3534 | KEYVAL_S- DAL0_5 |
| 0x0A00_3538 | KEYVAL_S- DAL0_6 |
| 0x0A00_353C | KEYVAL_S- DAL0_7 |
| 0x0A00_3540 | KEY- VAL_DAL_EL0 |
| 0x0A00_3544 | KEY- VAL_DAL_EL1 |
| 0x0A00_3548 | KEY- VAL_DAL_EL2 |
| 0x0A00_354C | KEY- VAL_DAL_EL3 |
| 0x0A00_3550 | KEY- VAL_DAL_EL4 |
| 0x0A00_3554 | KEY- VAL_DAL_EL5 |
| 0x0A00_3558 | KEY- VAL_DAL_EL6 |
| 0x0A00_355C | KEY- VAL_DAL_EL7 |


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

## 12.6. Fuse Summary: ROMCFG

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0x2FFF | Reserved | | | | | | | | | |
| 0x3000 | DAL | 7:0 | DAL_CPU[7:0] | | | | | | | |
| | | 15:8 | DAL_CPU[15:8] | | | | | | | |
| | | 23:16 | DAL_CPU[23:16] | | | | | | | |
| | | 31:24 | DAL_CPU[31:24] | | | | | | | |
| 0x3004 ... 0x33FF | Reserved | | | | | | | | | |
| 0x3400 | FRCFGBROMC | 7:0 | | | | | | | | BCRCDIS |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x3404 ... 0x3407 | Reserved | | | | | | | | | |
| 0x3408 | FRCFGMBIST | 7:0 | | | MOBB[1:0] | | MOBH[1:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 12.6.1. Device Access Level Register

**Name:** DAL
**Offset:** 0x3000
**Reset:** 0xFFFF_FF55
**Property:** Read Only

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DAL_CPU[31:24] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | DAL_CPU[23:16] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | DAL_CPU[15:8] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | DAL_CPU[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

**Bits 31:0 – DAL_CPU[31:0]** CPU Device Access Level

**Read Values:**
0xFFFF_FFFF = Un-programmed resolves to DAL0
0xFFFF_FF55 = DAL2
0xFF55_XXXX = DAL0
All other values = DAL0
**Write Values** (note, writes can be progressive):
To set DAL2 write 0xFFFF_FF55
To set DAL0 write 0xFF55_FFFF


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

### 12.6.2. Pre-Boot BROMC User Options Register

**Name:** FRCFGBROMC
**Offset:** 0x3400
**Reset:** 0x0A00_3400
**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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | BCRCDIS |
| Access | | | | | | | | R |
| Reset | | | | | | | | 1 |

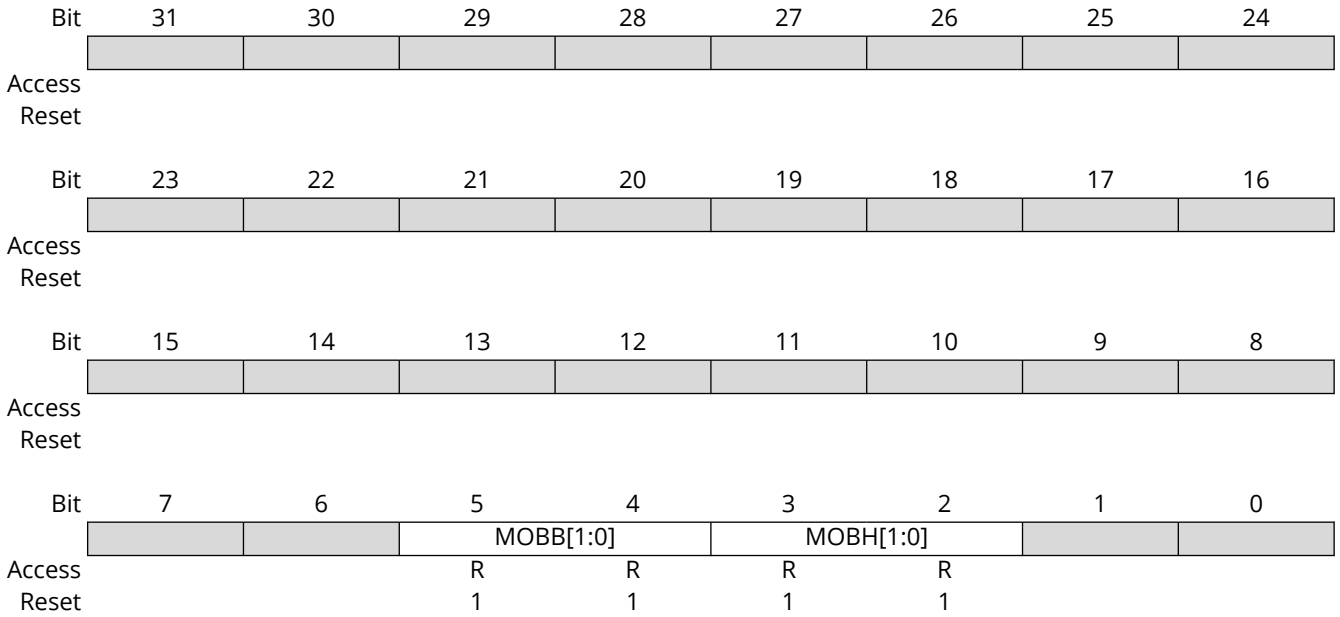
**Bit 0 – BCRCDIS** Boot ROM CRC Disable

| Value | Description |
|---|---|
| 0 | Boot ROM content is CRC verified by the ROM controller, status is reported to the DSU. On CRC failure, the controller blocks access to the ROM controller. |
| 1 | No CRC verification at Boot. |


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

### 12.6.3. Pre-boot MBIST User Options Register

**Name:** FRCFGMBIST
**Offset:** 0x3408
**Reset:** 0x0A00_3408
**Property:** Read Only




**Bits 5:4 – MOBB[1:0]** MBIST On Boot Enable for BACKUP/HIBERNATE Retaining Memory Groups

| Value | Name | Description |
|---|---|---|
| 11 | DISABLED | MOB Disabled |
| 10 | ENABLED_POR | MOB runs only after a POR |
| 01 | ENABLED_ANY | MOB runs after any non-BACKUP/HIBERNATE exit reset |
| 00 | Reserved | - |

**Bits 3:2 – MOBH[1:0]** MBIST On Boot Enable for HIBERNATE Only Retaining Memory Groups

| Value | Name | Description |
|---|---|---|
| 11 | DISABLED | MOB Disabled |
| 10 | ENABLED_POR | MOB runs only after a POR |
| 01 | ENABLED_ANY | MOB runs after any non-BACKUP/HIBERNATE exit reset |
| 00 | Reserved | - |


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

## 12.7. Fuse Summary: USERCFG

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0x07FF | Reserved | | | | | | | | | |
| 0x0800 | BOOT_FLAG | 7:0 | | RAM_INIT_EN B | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0804 ... 0x0817 | Reserved | | | | | | | | | |
| 0x0818 | BOOT_GPIOSEL | 7:0 | GPIOPORTSEL[2:0] | | | | GPIOPINSEL[3:0] | | | |
| | | 15:8 | | | FLTSLEWLIM[1:0] | | FLTODRAIN | | FLTPOL | GPIOPORTSE L[3] |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x081C ... 0x087B | Reserved | | | | | | | | | |
| 0x087C | FCW_CWP | 7:0 | | | | | RCWP | BC1WP | | BC1AWP |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | RCWPLOCK | BC1WPLOCK | | BC1AWPLOC K |
| | | 31:24 | | | | | | | | |
| 0x0880 | FCR_CRP | 7:0 | | | | | RCRP | BC1RP | | BC1ARP |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | RCRPLOCK | BC1RPLOCK | | BC1ARPLOCK |
| | | 31:24 | | | | | | | | |
| 0x0884 | ECCCTRL | 7:0 | | ECCUNLCK | ECCCTL[1:0] | | | | | |
| | | 15:8 | SECCNT[7:0] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0888 | SUPC_BRCFGUSMO R_BOR | 7:0 | | | | | | HYST_TRIP_VDDIO[1:0] | | HYST_BOR_V DDIO |
| | | 15:8 | | | | | | HYST_BOR_V DDREG | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x088C ... 0x088F | Reserved | | | | | | | | | |
| 0x0890 | SUPC_BRCFGUDSS MOR | 7:0 | | | | | | BOR_TRIP[1:0] | | BOR_HYST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0894 ... 0x0897 | Reserved | | | | | | | | | |
| 0x0898 | WDT_SETUP | 7:0 | ALWAYSON | RUNSTDBY | | | | WEN | ENABLE | |
| | | 15:8 | WINDOW[3:0] | | | | PER[3:0] | | | |
| | | 23:16 | | | | | EWOFFSET[3:0] | | | |
| | | 31:24 | | | | | | | | |


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

### 12.7.1. Boot Code Flags Register

**Name:** BOOT_FLAG
**Offset:** 0x800
**Reset:** 0xFFFF_FFF9
**Property:** R/P

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | RAM_INIT_ENB | | | | | | |
| Access | | R/P | | | | | | |
| Reset | | 1 | | | | | | |

**Bit 6 – RAM_INIT_ENB** RAM ECC Enable
When set to '1', ECC on RAM will be enabled.

| Value | Description |
|---|---|
| 0 | Disable ECC on RAM |
| 1 | Enable ECC on RAM |


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

### 12.7.2. Boot External Notification I/O Pin Register

**Name:** BOOT_GPIOSEL
**Offset:** 0x818
**Reset:** 0xFFFF_FFFF
**Property:** R/P

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | FLTSLEWLIM[1:0] | | FLTODRAIN | | FLTPOL | GPIOPORTSEL[3] |
| Access | | | R/P | R/P | R/P | | R/P | R/P |
| Reset | | | 1 | 1 | 1 | | 1 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | GPIOPORTSEL[2:0] | | | | GPIOPINSEL[3:0] | | | |
| Access | R/P | R/P | R/P | | R/P | R/P | R/P | R/P |
| Reset | 1 | 1 | 1 | | 1 | 1 | 1 | 1 |

**Bits 13:12 – FLTSLEWLIM[1:0]** Selects Driver Slew Rate Selection for Boot External Notification Signal

| Value | Description |
|---|---|
| 11 | Slew rate control disabled (fast rise/fall time operation) |
| 10 | Slew rate control enabled (4x slower) |
| 01 | Slew rate control enabled (8x slower) |
| 00 | Slew rate control enabled (12x slower) |

**Bit 11 – FLTODRAIN** Open Drain Control for Boot External Notification Signal

| Value | Description |
|---|---|
| 1 | The selected fault source normal drive |
| 0 | The selected fault source is open drain drive |

**Bit 9 – FLTPOL** Selects I/O Port for Boot External Notification Signal

| Value | Description |
|---|---|
| 1 | The selected fault source is active-low |
| 0 | The selected fault source is active-high |

**Bits 8:5 – GPIOPORTSEL[3:0]** Selects I/O Port for Boot External Notification Signal
Boot code verifies the existence of ports and limits programming of these values accordingly.
Any Other Value - Reserved, no Boot External Notification signal used

| Value | Description |
|---|---|
| 0011 | Port D |
| 0010 | Port C |


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

| Value | Description |
|---|---|
| 0001 | Port B |
| 0000 | Port A |

**Bits 3:0 – GPIOPINSEL[3:0]** Selects I/O Port Pin for Boot External Notification Signal
Boot code verifies the existence of ports and limits programming of these values accordingly.

| Value | Description |
|---|---|
| 11111 | Pin 31 |
| ... | |
| 00001 | Pin 1 |
| 00000 | Pin 0 |


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

### 12.7.3. CFM Page Write Protect Register

**Name:** FCW_CWP
**Offset:** 0x87C
**Reset:** 0x0000_0000
**Property:** R/P

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | RCWPLOCK | BC1WPLOCK | | BC1AWPLOCK |
| Access | | | | | R/S | R/S | | R/S |
| Reset | | | | | 0 | 0 | | 0 |

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | RCWP | BC1WP | | BC1AWP |
| Access | | | | | R/W | R/W | | R/W |
| Reset | | | | | 1 | 1 | | 1 |

**Bit 19 – RCWPLOCK** Panel 1 Lock Write Protection ROMCfg

| Value | Name | Description |
|---|---|---|
| 1 | LOCKED | The Lock and Write Protect bits for this Page are Locked and cannot be modified |
| 0 | UNLOCKED | The Lock and Write Protect bits for this Page are NOT Locked and can be modified |

**Bit 18 – BC1WPLOCK** Panel 1 Lock Write Protection BootCfg1

| Value | Name | Description |
|---|---|---|
| 1 | LOCKED | The Lock and Write Protect bits for this Page are Locked and cannot be modified |
| 0 | UNLOCKED | The Lock and Write Protect bits for this Page are NOT Locked and can be modified |

**Bit 16 – BC1AWPLOCK** Panel 1 Lock Write Protection BootCfg1A

| Value | Name | Description |
|---|---|---|
| 1 | LOCKED | The Lock and Write Protect bits for this Page are Locked and cannot be modified |
| 0 | UNLOCKED | The Lock and Write Protect bits for this Page are NOT Locked and can be modified |

**Bit 3 – RCWP** Panel 1 Write Protect ROMCfg

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Erase and Write Protection for this Page is Enabled |
| 0 | DISABLE | Erase and Write Protection for this Page is Disabled |

**Bit 2 – BC1WP** Panel 1 Write Protect BootCfg1


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

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Erase and Write Protection for this Page is Enabled |
| 0 | DISABLE | Erase and Write Protection for this Page is Disabled |

**Bit 0 – BC1AWP** Panel 1 Write Protect BootCfg1A

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Erase and Write Protection for this Page is Enabled |
| 0 | DISABLE | Erase and Write Protection for this Page is Disabled |


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

### 12.7.4. Status

**Name:** FCR_CRP
**Offset:** 0x880
**Reset:** 0x0000_0000
**Property:** R/P

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | RCRPLOCK | BC1RPLOCK | | BC1ARPLOCK |
| Access | | | | | R/P | R/P | | R/P |
| Reset | | | | | 1 | 1 | | 1 |

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | RCRP | BC1RP | | BC1ARP |
| Access | | | | | R/P | R/P | | R/P |
| Reset | | | | | 1 | 1 | | 1 |

**Bit 19 – RCRPLOCK** Panel 1 Lock Read Protection ROMCfg

| Value | Name | Description |
|---|---|---|
| 1 | LOCKED | The Lock and Read Protect bits for this Page are Locked and cannot be modified |
| 0 | UNLOCKED | The Lock and Read Protect bits for this Page are NOT Locked and can be modified |

**Bit 18 – BC1RPLOCK** Panel 1 Lock Read Protection BootCfg1

| Value | Name | Description |
|---|---|---|
| 1 | LOCKED | The Lock and Read Protect bits for this Page are Locked and cannot be modified |
| 0 | UNLOCKED | The Lock and Read Protect bits for this Page are NOT Locked and can be modified |

**Bit 16 – BC1ARPLOCK** Panel 1 Lock Read Protection BootCfg1A

| Value | Name | Description |
|---|---|---|
| 1 | LOCKED | The Lock and Read Protect bits for this Page are Locked and cannot be modified |
| 0 | UNLOCKED | The Lock and Read Protect bits for this Page are NOT Locked and can be modified |

**Bit 3 – RCRP** Panel 1 Read Protect ROMCfg

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Read Protection for this Page is Enabled |
| 0 | DISABLE | Read Protection for this Page is Disabled |

**Bit 2 – BC1RP** Panel 1 Read Protect BootCfg1

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Read Protection for this Page is Enabled |


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

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Read Protection for this Page is Disabled |

**Bit 0 – BC1ARP** Panel 1 Read Protect BootCfg1A

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Read Protection for this Page is Enabled |
| 0 | DISABLE | Read Protection for this Page is Disabled |


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

### 12.7.5. ECC Control REGISTER

**Name:** ECCCTRL
**Offset:** 0x884
**Reset:** 0x00000070
**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 |
|---|---|---|---|---|---|---|---|---|
| | SECCNT[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 |
|---|---|---|---|---|---|---|---|---|
| | | ECCUNLCK | ECCCTL[1:0] | | | | | |
| Access | | R/K | R/W | R/W | | | | |
| Reset | | 1 | 1 | 1 | | | | |

**Bits 15:8 – SECCNT[7:0]** SECCNT is the start value of an internal counter that decrements (by 1 per panel reporting an SEC) its count value each time an SEC occurs on a flash access (including the CTL[2:0] bits if in Dynamic ECC Mode). The internal counter stops decrementing at zero. If an SEC error occurs when the internal counter is zero, the SERR flag bit is set. This field counts all SEC errors and is not limited to SEC errors on unique addresses.

**Bit 6 – ECCUNLCK** The read value dictates the unlock state.

**Note:** This field can only be modified when ECCUNLCK=1.

The read value dictates the unlock state:

| Value | Name | Description |
|---|---|---|
| 1 | UNLOCKED | ECCUNLCK and ECCCTL[1:0] can be written |
| 0 | LOCKED | ECCUNLCK and ECCCTL[1:0] cannot be written |

**Bits 5:4 – ECCCTL[1:0]** Condition: {PFM_PNL_DATA_MSB==71} 10 (ECC) = Reserved - Resolves to STRICT 01 (DYNAMIC) = Reserved - Resolves to STRICT Note: This field can only be modified when ECCUNLCK=1
Restricts one or more NVMOPs:

| Value | Name | Description |
|---|---|---|
| 3 | DISABLE | ECC Writes with No Error Check Reads |
| 0 | STRICT | ECC Writes with ECC Reads |
| 3 | DISABLE | Dynamic Writes with No Error Check Reads |
| 2 | ECC | ECC Writes with Dynamic Reads (NVMOP = Single Program Operation disabled) |
| 1 | DYNAMIC | Dynamic Writes with Dynamic Reads |
| 0 | STRICT | ECC Writes with ECC Reads (NVMOP = Single Program Operation disabled) |


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

### 12.7.6. Boot ROM Configurable SMOR User CFG BOR Register

**Name:** SUPC_BRCFGUSMOR_BOR
**Offset:** 0x888
**Reset:** 0x0000_0000
**Property:** R/P

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | HYST_BOR_VDDREG | | |
| Access | | | | | | R/P | | |
| Reset | | | | | | 1 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | HYST_TRIP_VDDIO[1:0] | | HYST_BOR_VDDIO |
| Access | | | | | | R/P | R/P | R/P |
| Reset | | | | | | 1 | 1 | 1 |

**Bit 10 – HYST_BOR_VDDREG** ECC Disable

| Value | Name | Description |
|---|---|---|
| 0 | LOW | Low Hysteresis (default) |
| 1 | HIGH | High Hysteresis |

**Bits 2:1 – HYST_TRIP_VDDIO[1:0]** Nominal BOR Trip

| Value | Description |
|---|---|
| 11 | 2.93V |
| 10 | 2.64V |
| 01 | 2.2V |
| 00 | 1.68V (default) |

**Bit 0 – HYST_BOR_VDDIO** Nominal BOR Trip

| Value | Name | Description |
|---|---|---|
| 0 | LOW | Low Hysteresis (default) |
| 1 | HIGH | High Hysteresis |


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

### 12.7.7. Boot ROM Configurable DS SMOR USER CFG Register

**Name:** SUPC_BRCFGUDSSMOR
**Offset:** 0x890
**Reset:** 0x0000_0000
**Property:** R/P

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | BOR_TRIP[1:0] | | BOR_HYST |
| Access | | | | | | R/P | R/P | R/P |
| Reset | | | | | | 1 | 1 | 1 |

**Bits 2:1 – BOR_TRIP[1:0]** Nominal BOR Trip

| Value | Description |
|---|---|
| 11 | 2.8V |
| 10 | 2.55V |
| 01 | 2.12V |
| 00 | 1.61V (default) |

**Bit 0 – BOR_HYST**

| Value | Name | Description |
|---|---|---|
| 1 | HIGH | High Hysteresis |
| 0 | LOW | Low Hysteresis (default) |


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

### 12.7.8. Watchdog Timer Setup

**Name:** WDT_SETUP
**Offset:** 0x898
**Reset:** 0x00
**Property:** RW

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | EWOFFSET[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | WINDOW[3:0] | | | | PER[3:0] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ALWAYSON | RUNSTDBY | | | | WEN | ENABLE | |
| Access | RW | RW | | | | RW | RW | |
| Reset | 0 | 0 | | | | 0 | 0 | |

**Bits 19:16 – EWOFFSET[3:0]** Early Warning Interrupt Time Offset
These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt.

| 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 |

**Bits 15:12 – 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 1024 Hz CLK_WDT_OSC clock..

| Value | Name | Description |
|---|---|---|
| 0x0 | CYC8 | 8 clock cycles |
| 0x1 | CYC16 | 16 clock cycles |
| 0x2 | CYC32 | 32 clock cycles |
| 0x3 | CYC64 | 64 clock cycles |


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

| Value | Name | Description |
|---|---|---|
| 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 |

**Bits 11:8 – PER[3:0]** Time-Out Period

These bits determine the watchdog time-out period as a number of 1024 Hz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period.

| 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 |

**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.
Writing a '0' to this bit has no effect.

| 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 6 – RUNSTDBY** Run During Standby

This bit controls the behavior of the watchdog during Standby Sleep mode.

| Value | Description |
|---|---|
| 0 | The WDT is disabled during Standby sleep mode. |
| 1 | The WDT is enabled continues to operate during Standby sleep mode. |

**Bit 2 – WEN** Watchdog Timer Window Mode Enable

This bit enables Window mode.

| Value | Description |
|---|---|
| 0 | Window mode is disabled (normal operation). |
| 1 | Window mode is enabled. |

**Bit 1 – ENABLE** Enable

| Value | Description |
|---|---|
| 0 | The WDT is disabled. |
| 1 | The WDT is enabled. |


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

# 13. Multi-Channel RAM Controller (MCRAMC)

## 13.1. Overview

The MCRAMC provides memory controller functions as well as ECC and fault injection for the System RAM (DRM).

The memory controller provides up to 8 AHB ports which provide access the system RAM. Each port is guaranteed 100% bandwidth to the entire system RAM. However, there may be a latency penalties if more than one port attempts to access the same memory bank. The CPU master has exclusive access to AHB port 0 which is configured to a higher priority than other AHB ports. This configuration provides the CPU with the lowest overall latency to the system memory.

The memory controller also provides ECC and fault injection. The user can optional enable/disable ECC correction of read data. ECC values are always calculated and stored for writes. Both SEC and DED errors are logged and each event can generate an interrupt. On a SEC, the controller returns corrected data. On a DED, the controller generates a slave bus error. The MCRAMC implements up to four memory banks or channels. RAM must be distributed evenly across all channels.

## 13.2. Features

The following are key features of the MCRAMC:

Embedded Characteristics:

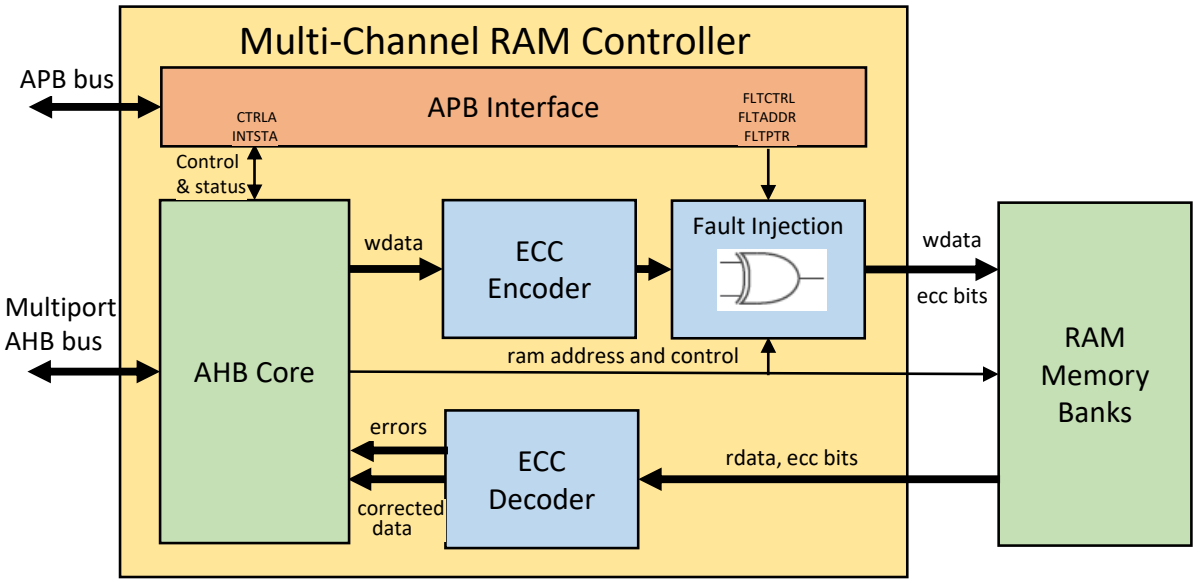
- Up to 8 AHB ports and up to 4 RAM channels provide for 100% bandwidth across all AHB ports
- Burst support
- 32-bit/64-bit memory access
- ECC support:
  - SEC DED
  - ECC on 32-bit/64-bit words
  - Byte write support with read modify write
  - AHB error response on DED
  - Fault Injection

## 13.3. Block Diagram

The following figure illustrates the block diagram of the MCRAMC.


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

**Figure 13-1.** MCRAMC Block Diagram




## 13.4. Functional Description

### 13.4.1. ECC SECDED - ECC Functionality

ECC = Error Correcting Code

SECDED = Single Error Correction Double Error Detection

The ECC decoding is enabled/disabled with the CTRLA.ENABLE bit. After reset the Rom code will copy the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse in the CTRLA.ENABLE bit.

The ECC feature can then be disabled and re-enabled using the same bit (for testing the ECC feature, for example, as detailed in the “ECC testing” chapter).

This disabling/enabling only applies to ECC decoding upon reads. The ECC encoding upon writes is always enabled and cannot be disabled.

Each 32-bit data word in SRAM is completed by 7 additional ECC bits, which are not accessible by the user.

Upon any 8-bit, 16-bit, or 32-bit write in the memory, the 7 ECC bits are computed and stored along with the data (the 8-bit or 16-bit writes are composed of an atomic read 32 bits, modify, write 32 bits).

Upon any 8-bit, 16-bit, or 32-bit read in the memory, if the ECC feature is disabled, then single or double errors are not detected, and not corrected. If the ECC feature is enabled, the ECC syndrome is computed on the related 32 data bits + 7 ECC bits.

Single-bit error read correction is performed on the fly with no penalty.

The MCRAMC writes back any corrected data into the RAM.

Simply reading the sensitive RAM content on a regular basis, for example from a DMA Host, prevents bit error accumulation. This is known as ECC memory scrubbing.

Double-bit error read detection triggers a bus error response from the MCRAMC, typically leading to a synchronous abort exception at the bus Host. This enables stopping of the bus Host access sequence precisely at the faulty address.


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

Upon a bus error response, the faulty address is registered in Error Capture Address Register.

If the corrupted RAM word laying at the 32-bit aligned address was originally provided from a non volatile memory, that is, CPU instruction, it can be safely copied again into the RAM before returning from the abort exception handler. If the abort exception is immediately re-entered, there are 'stuck-at' bits in the corrupted memory word, which requires applying self-patching code technics prior returning from the abort exception handler.

Another example is if the corrupted RAM word was part of a DMA buffer from a communication peripheral, the communication might be retried if the DMA master has support for transfer error response.

### 13.4.2. SRAM Initialization

After a reset the SRAM content (data and ECC) is random and the ECC feature may be enabled depending on the BOOTCFG1.BOOT_FLAG.RAM_INIT_ENB fuse value.

In case the ECC feature is enabled:

Any 32-bit write will initialize the data and the related ECC bits. However, 8-bit or 16-bit writes (which imply an internal read32/modify/write32) will probably trigger a single or double error on the internal 32-bit read (depending on the randomness of the 39 bits in memory). Consequently, the SRAM content must be initialized before it can be used.

The simplest option is to program a basic FOR loop filling the whole memory or to program a DMA transfer. The written data can take any value. However, care must be taken to only perform 32-bit writes in SRAM to access variables or DMA descriptors, and to not overwrite these data during the memory fill. The ECC bits will then be computed for each write and the memory will then be available for normal use.

### 13.4.3. ECC Testing

For ECC testing purpose, single-bit or double-bit faults can be injected during writes at a specific address. The address needs to be programmed in the MCRAMC Fault Injection Address Register (MCRAMC.FLTADDR), the bits to be flipped need to be programmed in the MCRAMC Fault Injection Pointer Register (MCRAMC.FLTPTR) and the fault injection needs to be enabled in the MCRAMC Fault Injection Control Register (MCRAMC.FLTCTRL).

Then the MCRAMC FLTCTRL register has to be read prior to the first write access to the RAM.

After fault injections, if ECC decoding is disabled in the MCRAMC Control Enable A Register, reading at a faulty address directly shows the faulty data bits, if any. Then ECC decoding can be enabled again and fault injection disabled, to read and check single error correction or double error detection. Then ECC decoding can be disabled again to read and check if a single error has been automatically corrected in the RAM array.

When ECC decoding is enabled in the MCRAMC Control Enable A Register, single-bit and double-bit ECC errors can be flagged in the MCRAMC INTSTA Interrupt Status register and can trigger an interrupt if enabled in the MCRAMC.INTENx Interrupt Enable register.

**Note:** This interrupt is an asynchronous CPU exception. It generally comes too late for system safe state recovery in case of a double-bit error.

The characteristics of a single-bit error are captured into the MCRAMC Error Capture registers and held stable until either the MCRAMC INTSTA.SER status bit is cleared by the software or a Double-bit error occurs.

The characteristics of a double-bit error are captured into the MCRAMC Error Capture registers and held stable until the MCRAMC INTSTA.DER status bit is cleared by the software.

The following constraints must be observed during the ECC testing process:


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

- After one or more writes to the MCRAMC user interface to change its configuration, a single read needs to be done from the MCRAMC FLTCTRL register, prior to performing any access to the RAM.
- When both ECC decoding and fault injection are enabled, no single-bit fault RAM word must be read at the RAM fault injection address, because memory correction write-back would inject a fault again.
- When both ECC decoding and fault injection are enabled, a double-bit fault RAM word at the RAM fault injection address must be overwritten only with a 32-bit wide access.
- When fault injection is enabled, the data bits to be flipped, as programmed in the MCRAMC Fault Injection Pointer Register, must always be part of the bytes modified by the write access to the RAM fault injection address. A simple way to ensure this is to restrict the write accesses to the RAM fault injection address to be 32-bit wide only.

**Note:** A double error condition will generate a hard fault which will supersede the double error interrupt. If identification and soft recovery is desired, this condition must be detected in the hard fault handler, allowing the function to exit and the ISR for the double error to be executed.

### 13.4.4. Throughput Optimization

The throughput of the MCRAMC can be optimized when the system bus clock frequency is in lower range by removing any wait-state in the access to the RAM.

At higher frequency a wait-state must be inserted at the start of a burst.

When the MCRAMC_CTRLA.WS = 1, a wait state is inserted.

Refer to the *Electrical Characteristics* chapter for the configuration of WS.

### 13.4.5. Privileged Access

When the MCRAMC_CTRLA.PRIV = 1, the user interface access is granted only if the read or write access is performed only if the privileged attribute is set.

## 13.5. Peripheral Dependencies

**Table 13-1.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| MCRAMC | 0x4402C000 | 34 : INT | **APB:** CLKMSK[0] MASK21 **AHB:** CLKMSK[0] MASK22 | --- | 16 STATUS0 PERID16 | | | PD_CORE_SW |


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

## 13.6. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | PRIV | ENABLE | SWRST |
| | | 15:8 | | | | | | | | WS |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | | | | | | | DERREN | SERREN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | INTENSET | 7:0 | | | | | | | DERREN | SERREN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | INTSTA | 7:0 | | | | | | | DERR | SERR |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | FLTCTRL | 7:0 | | | | | | | FLTEN | |
| | | 15:8 | | | FLTMD[1:0] | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | FLTPTR | 7:0 | FLT1PTR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | FLT2PTR[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | FLTADR | 7:0 | FLTADR[7:0] | | | | | | | |
| | | 15:8 | FLTADR[15:8] | | | | | | | |
| | | 23:16 | FLTADR[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | ERRCADR | 7:0 | ERCADR[7:0] | | | | | | | |
| | | 15:8 | ERCADR[15:8] | | | | | | | |
| | | 23:16 | ERCADR[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | ERRCPAR | 7:0 | ERCPAR[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | ERRCSYN | 7:0 | ERCSYN[7:0] | | | | | | | |
| | | 15:8 | ERR2 | ERR1 | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 13.6.1. Control Enable A Register

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000002
**Property:** RW

**Table 13-2.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | | | | | | WS |
| Reset | | | | | | | | 0 |

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

**Bit 8 – WS** Wait State Enable

**Bit 2 – PRIV** Privileged Access Enable

**Bit 1 – ENABLE** ECC Decoder Enable

| Value | Description |
|---|---|
| 0 | ECC decoding is disabled. |
| 1 | ECC decoding is enabled. |

**Bit 0 – SWRST** Software Reset

| Value | Description |
|---|---|
| 0 | No effect. |
| 1 | Reset the MCRAMC. A software-triggered hardware reset of the MCRAMC user interface is performed. |


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

### 13.6.2. Interrupt Enable Clear Register

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** R/W/K

**Table 13-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DERREN | SERREN |
| Access | | | | | | | | |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – DERREN** Double Bit Error Interrupt Enable, Clear

**Bit 0 – SERREN** Single Bit Error Interrupt Enable, Clear


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

### 13.6.3. Interrupt Enable Set Register

**Name:** INTENSET
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** R/W/S

**Table 13-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DERREN | SERREN |
| Access | | | | | | | | |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – DERREN** Double Bit Error Interrupt Enable, Set

**Bit 0 – SERREN** Single Bit Error Interrupt Enable, Set


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

### 13.6.4. Interrupt Status Register

**Name:** INTSTA
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** R/W/K

**Table 13-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DERR | SERR |
| Access | | | | | | | | |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – DERR** Double Bit Error

**Bit 0 – SERR** Single Bit Error


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

### 13.6.5. Fault Injection Control Register

**Name:** FLTCTRL
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** RW

**Table 13-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | FLTMD[1:0] | | | | | |
| Reset | | | 0 | 0 | | | | |

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

**Bits 13:12 – FLTMD[1:0]** Fault Injection Mode

**Bit 1 – FLTEN** Fault Injection Enabled

| Value | Description |
|---|---|
| 0 | Disables fault injection. |
| 1 | Enables fault injection at FLTADR address offset as selected by FLTMD and FLTxPTR. |


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

### 13.6.6. Fault Injection Pointer Register

**Name:** FLTPTR
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** RW

**Table 13-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FLT2PTR[7:0] | | | | | | | |
| Access | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | FLT1PTR[7:0] | | | | | | | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:16 – FLT2PTR[7:0]** Double Fault Injection Bit Pointer

**Bits 7:0 – FLT1PTR[7:0]** Single Fault Injection Bit Pointer


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

### 13.6.7. Fault Injection Address Register

**Name:** FLTADR
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** RW

**Table 13-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

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

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

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

**Bits 23:0 – FLTADR[23:0]** Fault Address Offset


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

### 13.6.8. Error Capture Address Register

**Name:** ERRCADR
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** R

**Table 13-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | ERCADR[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 |
|---|---|---|---|---|---|---|---|---|
| | ERCADR[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 |
|---|---|---|---|---|---|---|---|---|
| | ERCADR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 23:0 – ERCADR[23:0]** ECC SECDED Error Capture Address


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

### 13.6.9. Error Capture Parity Register

**Name:** ERRCPAR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** R

**Table 13-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | ERCPAR[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – ERCPAR[7:0]** ECC SECDED Error Capture Parity


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

### 13.6.10. Error Capture Syndrome Register

**Name:** ERRCSYN
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** R

**Table 13-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | ERR2 | ERR1 | | | | | | |
| Access | R | R | | | | | | |
| Reset | 0 | 0 | | | | | | |

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

**Bit 15 – ERR2** ECC Double Bit Error

| Value | Description |
|---|---|
| 0 | Not a Double bit error. |
| 1 | Double bit error. |

**Bit 14 – ERR1** ECC Single Bit Error

**Bits 7:0 – ERCSYN[7:0]** ECC SECDED Error Capture Syndrome

| Value | Description |
|---|---|
| 0 | Not a Single bit error. |
| 1 | Single bit error. |


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

# 14. Peripheral Access Controller (PAC)

## 14.1. Overview

The Peripheral Access Controller provides an interface for the locking and unlocking of predefined peripheral registers within the device. It controls reporting of 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. If access errors are reported by a bus errors (aka bus faults). The PAC module also reports errors occurring at the client bus level, when an access to a non-existing address is detected.

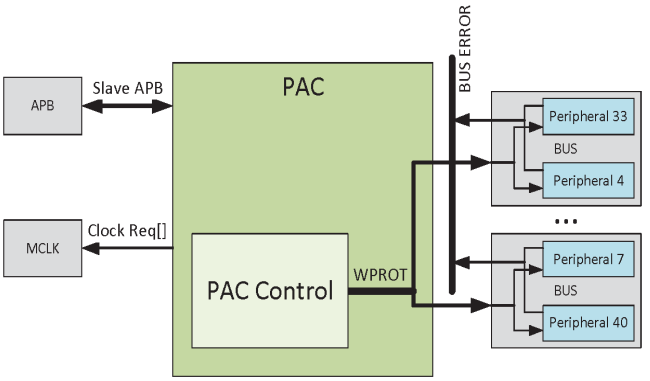
## 14.2. Features

The following are key features of the PAC module:

- Programmable write protect bit for each peripheral that supports it
- Interrupt generation on access errors to peripherals
  - Write protect error
  - Unused Address
  - Access when clock synchronization is active
  - Access when software reset is on-going
  - Access to reserved memory regions

## 14.3. Block Diagram

**Figure 14-1.** PAC Block Diagram




## 14.4. Debug Operation

When the CPU is halted in Debug mode, write protection of all peripherals is disabled and the PAC continues normal operation.

## 14.5. Register Access Protection

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

- Write Control (WRCTRL) register

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.

PAC write protection does not apply to accesses through an external debugger.

## 14.6. Functional Description

### 14.6.1. Basic Operation

#### 14.6.1.1. Operations

The PAC module allows the user to set, clear, or lock the write protected status of any peripherals. If a peripheral access violation occurs, the PAC reports it as an error on the client bus level. The corresponding Peripheral Write Control Status n register (STATUSx) gives the state of the write protection for all peripherals.

#### 14.6.1.2. 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 data sheet 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.
- 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.

#### 14.6.1.3. 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 a unique identifier corresponding to a peripheral (The value corresponding to each peripheral can be found in the peripheral dependency table) . 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.

Any peripheral access control status can be read from the corresponding STATUSn register and its lock state from LOCKn register.

#### 14.6.1.4. Write Access Protection Management Errors

Only word-wise (or register-wise) writes to the WRCTRL register will change the access protection. Other types of accesses will have no effect and will cause a PAC write access error.

The PAC also generates an in-band error in the following conditions:

- If a peripheral is write protected and a subsequent set protection or set and lock operation is detected then the PAC returns an error.
- If a peripheral is not protected and a subsequent clear protection operation is detected then the PAC returns an error.
- If a peripheral is write protected and locked and either a clear protection or set protection operation is detected, then the PAC returns an error.


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

This can be used to ensure that the application follows any write protect with an unprotect and any unprotect with a protect. In applications where the write protect state of a peripheral is manipulated in several contexts, care should be taken so that the contexts do not interfere with each other.

### 14.6.2. 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.

## 14.7. Peripheral Dependencies

**Table 14-1.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| PAC | 0x44022000 | | **APB:**<br>CLKMSK[0]<br>MASK14 | --- | 14<br>STATUS0<br>PERID14 | | | PD_CORE_SW |


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

## 14.8. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | PRIV | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | WRCTRL | 7:0 | PERID[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | KEY[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | EVCTRL | 7:0 | | | | | | | | ERREO |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | STATUS0 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| 0x44 | STATUS1 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| 0x48 | STATUS2 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| 0x4C | STATUS3 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| 0x50 | LOCK0 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| 0x54 | LOCK1 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| 0x58 | LOCK2 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| 0x5C | LOCK3 | 7:0 | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| | | 15:8 | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| | | 23:16 | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| | | 31:24 | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |


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

### 14.8.1. Control A Register

**Name:** CTRLA
**Offset:** 0x0
**Reset:** 0x00000000
**Property:** R/W

**Table 14-2.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | PRIV | | |
| Access | | | | | | R/W | | |
| Reset | | | | | | 0 | | |

**Bit 2 – PRIV** Privileged Access Only

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Registers only accessible in privileged accesses. |
| 0 | DISABLE | Registers accessible in privileged and unprivileged accesses. |


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

### 14.8.2. Write Control Register

**Name:** WRCTRL
**Offset:** 0x4
**Reset:** 0x00000000
**Property:** W

**Table 14-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | KEY[7:0] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PERID[7:0] | | | | | | | |
| Access | W | W | W | W | W | W | W | W |
| 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 |
|---|---|---|
| 3 | LOCK | Set and Lock the write protection state of the peripheral until the next reset |
| 2 | SET | Set the peripheral write protection |
| 1 | CLEAR | Clear the peripheral write protection |
| 0 | OFF | No Action |

**Bits 7:0 – PERID[7:0]** Peripheral ID
The PERID represents the peripheral whose control is changed using the WRCTRL.KEY.
The value corresponding to each peripheral can be found in the peripheral dependency table.


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

### 14.8.3. Event Control Register

**Name:** EVCTRL
**Offset:** 0x8
**Reset:** 0x00000000
**Property:** R/W

**Table 14-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | ERREO |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – ERREO** Error Event Output

This bit indicates if the Peripheral Access Error Event Output is enabled or disabled. When enabled, the event is asserted while any INTFLAGn bit is set.

| Value | Description |
|---|---|
| 1 | Peripheral Access Error Event Output is enabled. |
| 0 | Peripheral Access Error Event Output is disabled. |


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

### 14.8.4. Peripheral Status n Register

**Name:** STATUS
**Offset:** 0x40 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** R

Reading the STATUSn register returns the peripheral write protection status of the indicated peripherals:

0: Peripheral is not write protected.

1: Peripheral is write protected.

**Note:** The PERIDx value corresponding to each peripheral can be found in the peripheral dependency table.

**Table 14-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| 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 – PERID** PERID Write Protection Status

| Value | Description |
|---|---|
| 1 | Write Protection enabled |
| 0 | Write Protection disabled |


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

### 14.8.5. Peripheral Lock n Register

**Name:** LOCK
**Offset:** 0x50 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** R

**Table 14-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PERID31 | PERID30 | PERID29 | PERID28 | PERID27 | PERID26 | PERID25 | PERID24 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PERID23 | PERID22 | PERID21 | PERID20 | PERID19 | PERID18 | PERID17 | PERID16 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PERID15 | PERID14 | PERID13 | PERID12 | PERID11 | PERID10 | PERID9 | PERID8 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PERID7 | PERID6 | PERID5 | PERID4 | PERID3 | PERID2 | PERID1 | PERID0 |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| 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 – PERID** Period ID

PERID Write Protection Lock Status

| Value | Description |
|---|---|
| 1 | Write Protection Locked |
| 0 | Write Protection Unlocked |


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

# 15. Device Service Unit (DSU)

## 15.1. Overview

The Device Service Unit (DSU) provides multiple functions which include debugger access restrictions, device variant configuration, and Microchip device ID. The DSU is responsible for enabling, disabling, or restricting plugging detection and debugger access into the device. It also provides the Arm CoreSight Debug ROM Table and a communication mailbox for ROM services.

## 15.2. Features

The following are key features of the DSU module:

- Arm® CoreSight™ compliant device identification
- Microchip device identification
- CPU reset extension
- Debugger probe detection (Cold-Plugging and Hot-Plugging)
- Chip-Erase command and status
- Two Debug Communication Channels (DCC[0,1] with DMA signaling)
- Two Boot Communication Channels (BCC[0,1])
- Debug access port security filter

## 15.3. Signal Description

The DSU uses these signals to function.

| Signal Name | Type | Description |
|---|---|---|
| $\overline{\text{RESET}}$ | Digital Input | External Reset |
| SWCLK/TCK | Digital Input | Serial Wire Debug (SWD) clock / JTAG Test Clock |
| SWDIO/TMS | Digital I/O | Serial Wire Debug (SWD) bidirectional data pin / JTAG Test Mode Select |
| TCK | Digital Input | JTAG Test Clock |
| TMS | Digital Input | JTAG Test Mode Select |
| TDI | Digital Input | JTAG Test Data In |
| TDO | Digital Output | JTAG Test Data Out |

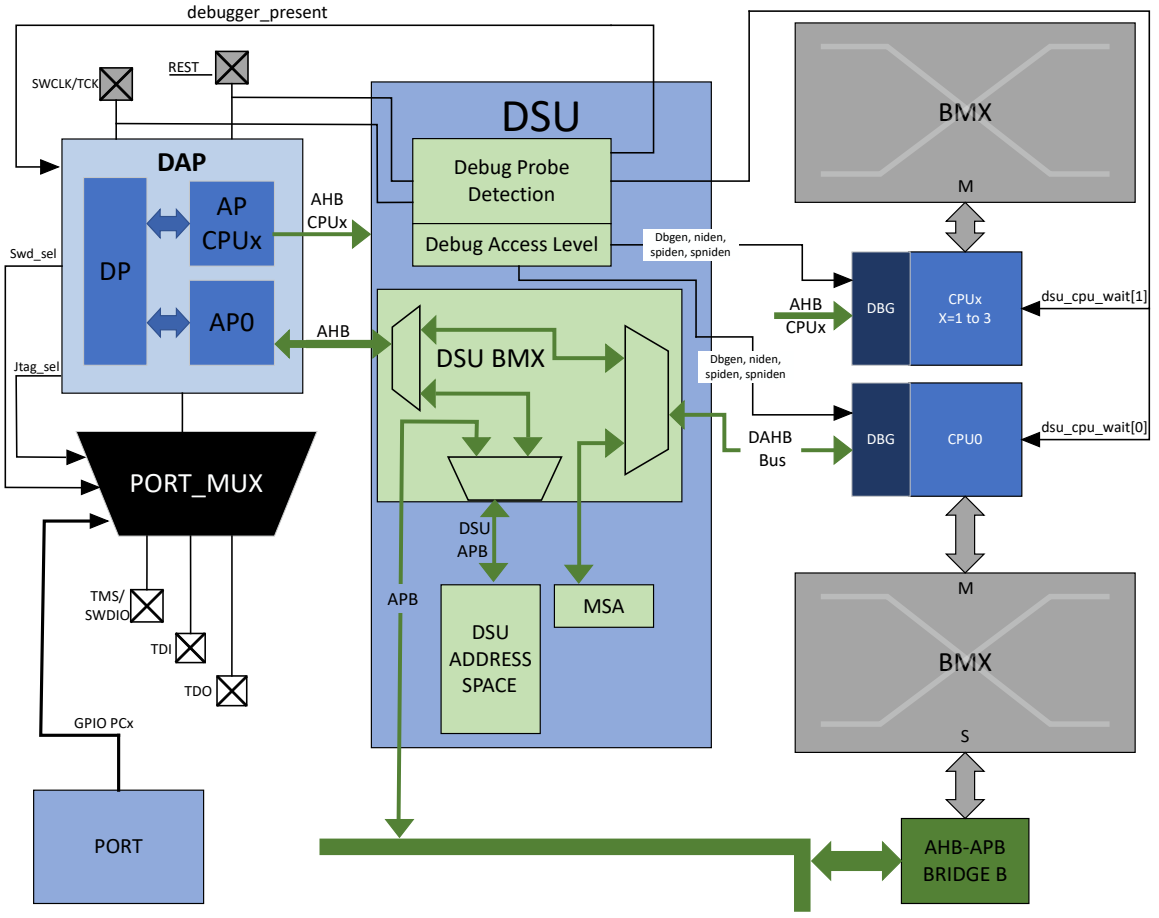
**Notes:**

1. By default the Debug Port (DP) starts in JTAG mode after Power-on Reset, but it can switch to Single Wire Debug (SWD) mode using the Arm JTAG to SWD switching sequence.
2. Any I/O pins being used for SWD/JTAG functions should not be configured for Open Drain operation.


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

## 15.4. DSU Block Diagram

**Figure 15-1.** Block Diagram




## 15.5. Indexing

The DSU can service up to two CPUs: CPU0 is the device's core CPU. Any other CPU indexed higher then 0 is an optional co-processor.

## 15.6. Debug Operation

The DSU monitors the TCK/SWCLK and RESETN pins to determine if a debugger is attached and a debug session can be initiated. The DSU supports detection of the debugger at reset, Cold-Plugging, and detection of the debugger when the device is out of reset, Hot-Plugging.

### 15.6.1. Principle of Operation

The DSU monitors the presence of a debugger to control appropriately on-chip resources when a debugger is attached to the device. The following are two attachment methods with different properties discussed later:

- Cold-plugging: Attachment when external reset is asserted.
- Hot-plugging: Attachment when external reset is not asserted.

A debugger presence has several effects on the device:

- Assigns JTAG or SWD pads to the Debug Port (DP) by controlling the port mux


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

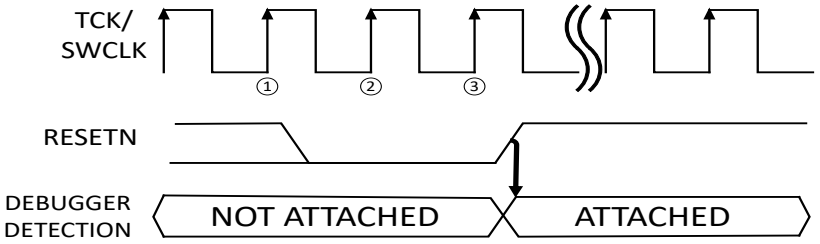
- Enables Access Ports
- Extends the device CPU's Boot ROM and reset phases

The DSU also implements Boot Communication Channels to enable communication with the device's Boot ROM. For additional information on the Arm debug components, refer to the Arm "Debug Interface v5 Architecture Specification" document.

### 15.6.2. Cold Plugging

Cold-Plugging detection occurs when the external reset is asserted for three consecutive TCK/SWCLK rising edges. A Cold-Plugging debug sessions terminates on POR or a brown out reset. The device TCK/SWCLK pad applies an internal pull-up to avoid false detection of a connection when this pin is left unconnected on the application board. Cold-plugging detection is available once pads are correctly powered, therefore after a power-on-reset. The external reset should be held asserted beyond the power-reset time for proper detection of cold-plugging.

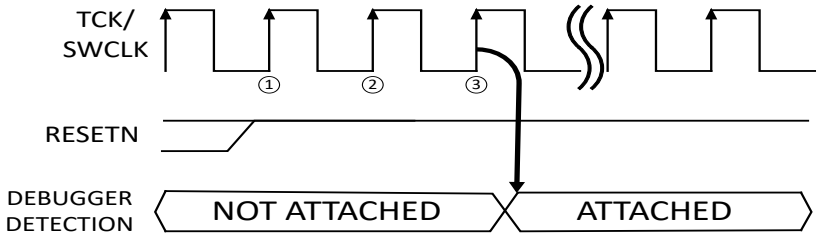
**Figure 15-2.** Cold-Plugging Timing Diagram




### 15.6.3. Hot-Plugging

Hot-plugging detection occurs when the external reset pin is not asserted and at least three TCK/SWCLK rising edges are detected. A hot-plugging debug sessions terminates only on a Brown-out-Reset . Hot-plugging is possible only if the device PORT MUX selects the TCK/SWCLK function. Availability of the hot-plugging feature can be determined from the STATUSB.HPS Hot-Plugging status bit. The figure below illustrates when hot-plugging is available based on debug access level. If the debugger IOs are programed for a user function besides debug, STATUSB.HPS = 0 regardless of the debug access level.

**Figure 15-3.** Hot-Plugging Detection Timing Diagram




### 15.6.4. Debugger Probe Detection

#### 15.6.4.1. Debugger Probe Detection Conditions and Effects

The presence of a debugger probe is detected when either Hot-Plugging or Cold-Plugging is detected .

Effects of a debugger detection are as follows:

- PORT MUX JTAG or SWD functions are selected depending on the Debug Port (DP) selected protocol. These functions have a higher priority than the GPIO function controlled by the PORT module. User code cannot reclaim the pads that are claimed by the DP from that point on by configuring the PORT. The number of pads claimed by the DP depends on the DP selected


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

protocol (JTAG or SWD). The DP starts in JTAG mode after a power-reset (TCK, TMS, TDI, TDO are claimed in this case) but can switch to SWD mode using the Arm JTAG to SWD switching sequence (the SWCLK and SWDIO pads only are claimed by the DP in this case).

- Access ports are enabled. Trying to access an access port register from the DP when it is disabled returns a DP fault (sets the DP sticky error bit).
- The Debugger Present bit of the Status B register (STATUSB.DBGPRES) reads one
- Extends the CPU reset (only with cold-plugging)

**Note:** Once in SWD mode it is possible to switch back to JTAG mode using the Arm SWD to JTAG switching sequence.

### 15.6.5. CPU Reset Extension

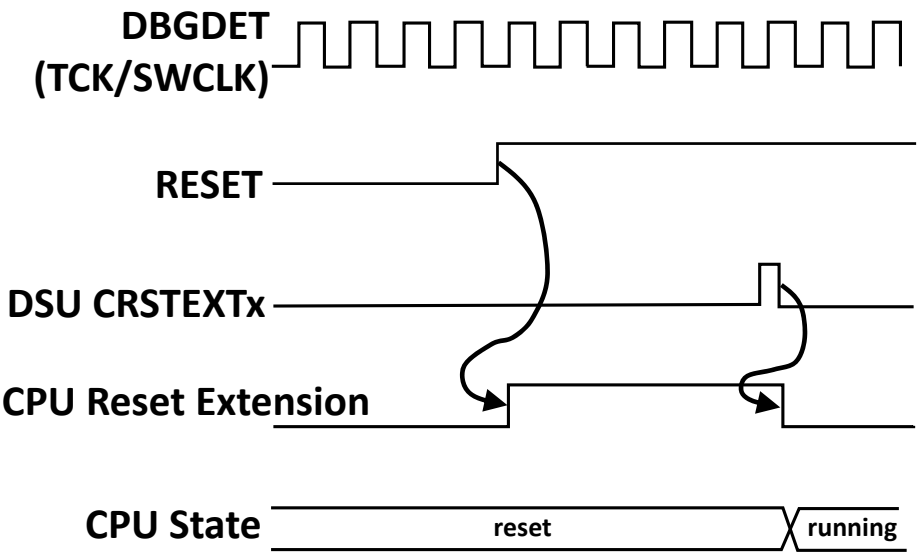
CPU reset extension refers to the extension of the reset phase of the device's CPU cores when a debugger connects with the cold-plugging procedure. This ensures that when a debugger connects:

- The CPU is not executing code at startup
- The device is in a safe and in a known state (reset)

For security reasons, the hot-plugging method is not allowed when CPU0.DAL = 0. Cold-plugging must be used in this case.

For each CPU held in the reset extension phase, the CPU Reset Extension bits of the Status A register (STATUSA.CRSTEXTx) reads one. To release CPUx from reset, write a '1' in STATUSA.CRSTEXTx. STATUSA.CRSTEXTx will then be cleared. More than one CPU can be released at a time (writing a '0' to STATUSA.CRSTEXTx has no effect). Releasing the CPU reset extension is possible for all DAL.CPUx levels. Each CPU then executes its Boot ROM. The host CPU (CPU 0) Boot ROM implements security checks at startup. It is not possible to access the bus system until the Boot ROM has performed these security checks.

**Figure 15-4.** Typical CPU Reset Extension Set and Clear Timing Diagram




### 15.6.6. JTAG/SWD PORT BEHAVIOR

Once the debugger is attached, the DSU takes ownership of the JTAG port pins to prevent application code from reclaiming them. If the debugger switches to SWD mode using the Arm JTAG to SWD switching sequence, the unused JTAG I/Os are released, claiming the SWCLK and SWDIO I/Os


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

only. Once in SWD mode, the debugger can switch back to JTAG mode using the Arm SWD to JTAG switching sequence at which point the hardware will claim all the JTAG I/Os again.

**Note:** Releasing and reclaiming of JTAG ports are handled outside of the DSU and the DSU is unaware of this operation.

### 15.6.7. Multi-Processor Support

The DSU can support multiple CPU cores. If there is a single-core, it will be referred to as CPU0. Additional cores will be indexed as CPU1, CPU2, and so on. Each CPU connects a different bus system. The DAP implements one Memory Access Port (MEM-AP) per CPU and allows debug tools to access each CPU's bus system. Debug tools must select the correct MEM-AP at the debug port level depending on the targeted bus system.

- MEM-AP0 is disabled (by decreasing order of priority) when:
  - CPU0.DAL = 0 and no cold-plugging detected.

Once enabled, MEM-AP0 always has access to the DSU external space to enable communications between the device and external tools.

- MEM-AP1 is disabled when:
  - When STATUSB.DBGPRES = 0
  - when their respective DAL.CPU1 level is not equal to 2.

### 15.6.8. Boot ROM Phase Extension

On the cold-plugging attachment, the DSU sets the STATUSA.BREXTx bit which notifies the CPUx's Boot ROM to wait until BREXTx clears before continuing the boot flow, where 'x' ranges from 0 to the number of CPUs. The debugger can write a '1' to the BREXTx bit to clear the bit and end this phase extension.

### 15.6.9. Debugger Security

The DSU provides Debug Access Level (DAL) mechanism which disable or limit the debugger access into the device. The DAL settings are programed through the fuse configuration bits provided by the device.

The DSU DAL feature provides three levels of access as provided in the table below:

**Table 15-1.** Debugger Access Level Through DAP

| DAL Setting | DSU Address Space | | System Memory Regions [1] | | Hot Plugging[3] | Cold Plugging |
|---|---|---|---|---|---|---|
| | External | Internal | Secure[2] | Non-Secure | | |
| DAL 0 | X | - | - | - | - | X |
| DAL 1[1][2] | X | - | - | X | X | X |
| DAL 2 | X | X | X | X | X | X |

**Notes:**

1. Only available on SG devices. Secure requests are converted to non-secure request in the CPU.
2. Only available on TrustZone devices.
3. Refer to the section on hot-plugging for further restrictions on availability of hot-plugging.

If the debugger attempts to access a device region which is not permitted by the current DAL setting, a bus error is returned to the CoreSight AP hardware which results in the assertion of the Arm DP sticky error bit.

In DAL0 mode, debugger breakpoints are disabled.


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

### 15.6.10. Debug Communication Channels

The Debug Communication Channels (DCC0 and DCC1) consist of a pair of registers with associated handshake logic, accessible by both CPU and debugger with no security restriction. These registers are used to exchange data between the device and a debugger. This enables the user to build a custom debug protocol using only these registers.

The DCC0 and DCC1 registers are always accessible from the external address space. They are used to communicate with the Boot ROM after a cold-plugging procedure.

Two Debug Communication Channel status bits in the STATUSB register (STATUSB.DCCDx) indicate that a new value has been written into DCC0 or DCC1. The DCC0D and DCC1D bits are located in the STATUSB register. They are automatically set by hardware and cleared on read.

**Note:** The DCC0 and DCC1 registers are shared with the Boot Communication Channels (BCC0 and BCC1) registers, therefore mixing DCC and BCC communication is not recommended.

Each DCC register also has a DMA request associated with it which can either be configured to assert when the DCCn register is full or empty. The DMA can be configured to either read from or write to the DCCn channel on a DMA request, thus allowing the DMA to transfer data into and out of the debugger through the DCCn registers.

#### 15.6.10.1. Boot Communication Channels

The CPU0 Boot ROM controller is capable of performing CRC verification of the ROM content. The result of the CRC check is reported by the STATUSA.ROMCRC status field which is always accessible by the debugger regardless of the DAL security level.

Since the CPU0 Boot ROM is the device root of trust, if the ROM content fails CRC verification, the ROM code never executes, the effective DAL setting remains at DAL0 for all CPU interfaces.

## 15.7. Programming

For each bus system, programming the Flash or RAM memories is only possible when the debugger access level is sufficient to access the desired resource. If MEM-APx is selected by the DP and if DAL.CPU0 is equal to:

- 0x2: debugger can access all areas in the CPU bus system.
- 0x0: If the debugger targets the MEM-AP0 then it can only access the DSU external address space making it possible to communicate with the Boot ROM after reset. If debugger targets the MEM-AP1 then all transactions results in a DAP fault.

A typical programming procedure, when DAL.CPU0 = 0x2, is presented below:

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. The system continues to be held in this static state until all the internally regulated supplies have reached a safe operating state.
2. The Power Manager (PM) starts, clocks are switched on (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 generates at least 3 SWCLK clock cycles while $\overline{\text{RESET}}$ is asserted. $\overline{\text{RESET}}$ is then 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. CPU 0 executes its Boot ROM.
6. It is recommended to issue a chip-erase (supported by the Boot ROM) to ensure that the Flash is fully erased prior to programming.
7. If the operation issued above was accepted and has completed successfully then DAL.CPU0 equals 0x2, therefore programming is available through the MEM-AP 0.


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

8. After the operation is completed, the chip can be restarted either by asserting $\overline{\text{RESET}}$, toggling power, or sending a command to the Boot ROM to jump to the NVM code. Make sure that the SWCLK clock is stopped while asserting $\overline{\text{RESET}}$ to prevent entering again the cold-plugging procedure.

## 15.8. Device Identification

The DSU provides both the Arm CoreSight Identification registers and the Microchip Device ID register, both are accessible by the debugger interface and APB client interface.

### 15.8.1. 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:

**Figure 15-5.** Conceptual 64-bit Peripheral ID

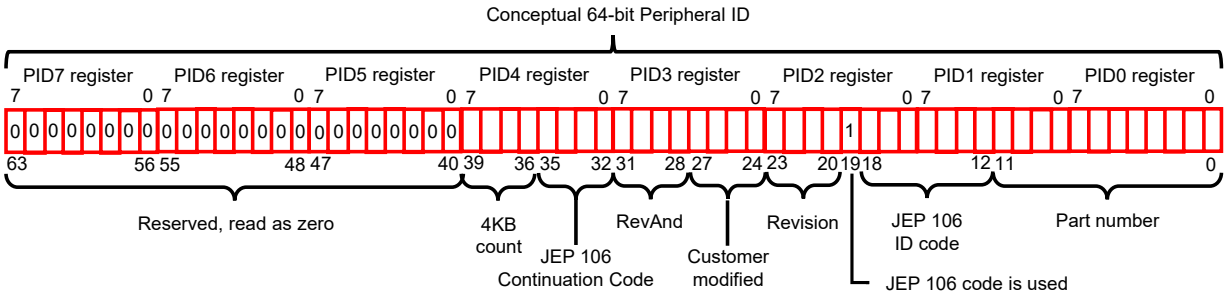



**Table 15-2.** Conceptual 64-Bit Peripheral ID Bit Descriptions

| Field | Size | Description | Location |
|---|---|---|---|
| JEP-106 CC code | 4 | Microchip Continuation code: 0x0 | PID4 |
| JEP-106 ID code | 7 | Device ID: 0x29 | PID1+PID2 |
| 4KB count | 4 | Indicates that the CoreSight component is a ROM: 0x0 | PID4 |
| Rev And | 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. this indicates that device identification can be completed by reading the Device Identification register (DID). | PID0+PID1 |
| REVISION | 4 | DSU revision (starts at 0x0 and increments by 1 at both major and minor revisions). Identifies DSU identification method variants. | PID2 |

For additional information, refer to the “Arm Debug Interface Version 5 Architecture Specification”.

### 15.8.2. Chip Identification Method

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

- Chip revision
- Part number


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

## 15.9. Peripheral Dependencies

**Table 15-3.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| DSU | 0x44000000 | | **AHB:**<br>CLKMSK[0]<br>MASK0 | --- | 0<br>STATUS0<br>PERID0 | 1 : DCC0<br>2 : DCC1 | | PD_CORE_SW |


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

## 15.10. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | PRIV | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CTRLB | 7:0 | | | | | | | DCCDMALVL1 | DCCDMALVL0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 ... 0xFF | Reserved | | | | | | | | | |
| 0x0100 | CTRLC | 7:0 | | | | | | | | DBGRW0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0104 | STATUSA | 7:0 | ROMCRC[1:0] | | | | | BERR | | |
| | | 15:8 | | | | | | | | CRSTEXT0 |
| | | 23:16 | | | | | | | | BREXT0 |
| | | 31:24 | | | | | | | | |
| 0x0108 | STATUSB | 7:0 | | | | | DCCD1 | DCCD0 | BCCD1 | BCCD0 |
| | | 15:8 | | | | | | HPS | | DBGPRES |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x010C ... 0x010F | Reserved | | | | | | | | | |
| 0x0110 | BCC0 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x0114 | BCC1 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x0118 | DCC0 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x011C | DCC1 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x0120 | DID | 7:0 | MANID[6:0] | | | | | | | FV |
| | | 15:8 | PNDID[3:0] | | | | MANID[10:7] | | | |
| | | 23:16 | PNMID[3:0] | | | | PNDID[7:4] | | | |
| | | 31:24 | VER[3:0] | | | | PNMID[7:4] | | | |
| 0x0124 | DAL | 7:0 | CPU3[1:0] | | CPU2[1:0] | | CPU1[1:0] | | CPU0[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0128 ... 0x0FFF | Reserved | | | | | | | | | |
| 0x1000 | ENTRY0 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x1004 | ENTRY1 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x1008 | ENTRY2 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x100C | ENTRY3 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x1010 | ENTRY4 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x1014 | ENTRY5 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x1018 | ENTRY6 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x101C | ENTRY7 | 7:0 | | | | | | | FMT | EPRES |
| | | 15:8 | ADDOFF[3:0] | | | | | | | |
| | | 23:16 | ADDOFF[11:4] | | | | | | | |
| | | 31:24 | ADDOFF[19:12] | | | | | | | |
| 0x1020 ... 0x1FCB | Reserved | | | | | | | | | |
| 0x1FCC | MEMTYPE | 7:0 | | | | | | | | SYSMEM |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FD0 | PID4 | 7:0 | SIZE[3:0] | | | | DES2[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FD4 | PID5 | 7:0 | RES0[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FD8 | PID6 | 7:0 | RES0[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FDC | PID7 | 7:0 | RES0[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FE0 | PID0 | 7:0 | PART0[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FE4 | PID1 | 7:0 | DES0[3:0] | | | | PART1[3:0] | | | |
| | | 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x1FE8 | PID2 | 7:0 | REVISION[3:0] | | | | JEDEC | DES1[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FEC | PID3 | 7:0 | REVAND[3:0] | | | | CMOD[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FF0 | CID0 | 7:0 | PRMBL0[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FF4 | CID1 | 7:0 | CCLASS[3:0] | | | | PRMBL1[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FF8 | CID2 | 7:0 | PRMBL2[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1FFC | CID3 | 7:0 | PRMBL3[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 15.10.1. Control A REGISTER

**Name:** CTRLA
**Offset:** 0x0
**Reset:** 0x00000000
**Property:** R/W

**Table 15-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | PRIV | | |
| Access | | | | | | R/W | | |
| Reset | | | | | | 0 | | |

**Bit 2 – PRIV**

DAPAHB access to DSU Internal Address Space registers are also affected by PRIV setting. If PRIV=1, non-privileged DAP AHB accesses to DSU Internal Address Space registers return an AHB Client bus error. DSU External address space and DSU CoreSight ROM Address Space are not affected by the PRIV bit.

| Value | Description |
|---|---|
| 1 | Internal Address Space registers only accessible in privileged mode. |
| 0 | Internal Address Space registers accessible in privileged and unprivileged modes. |


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

### 15.10.2. Control B REGISTER

**Name:** CTRLB
**Offset:** 0x4
**Reset:** 0x00000000
**Property:** R/W

**Table 15-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | DCCDMALVL[1] | DCCDMALVL[0] |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 0, 1 – DCCDMALVL** DMA Trigger n Level

| Value | Name | Description |
|---|---|---|
| 1 | FULL | Trigger n rises when DCCn is written and falls when it is read |
| 0 | EMPTY | Trigger n rises when DCCn is read and falls when it is written |


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

### 15.10.3. Control C REGISTER

**Name:** CTRLC
**Offset:** 0x100
**Reset:** 0x00000000
**Property:** R/W

**Table 15-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRW0 |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRW**

| Value | Description |
|---|---|
| 1 | Debugger accesses through CPUn do not carry the debug attribute (i.e debugger reads are destructive). |
| 0 | Allow debugger accesses through CPUn to carry the debug attribute (i.e debugger reads are non-destructive). |


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

### 15.10.4. Status A REGISTER

**Name:** STATUSA
**Offset:** 0x104
**Reset:** 0x00000000
**Property:** R/K

**Table 15-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | HS/HC/R/K |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | HS/HC/R/K |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Access | HS/HC/R | HS/HC/R | | | | HS/R/K | | |
| Reset | 0 | 0 | | | | 0 | | |

**Bit 16 – BREXT**

| Value | Description |
|---|---|
| 1 | Boot ROM Phase Extension for CPUx. Set on Cold-Plugging detection. |
| 0 | Boot ROM Phase Extension released for CPUx. |

**Bit 8 – CRSTEXT**

| Value | Description |
|---|---|
| 1 | CPUx held in reset extension. Set on Cold-Plugging detection. |
| 0 | CPUx released from reset extension. |

**Bits 7:6 – ROMCRC[1:0]** Boot ROM CRC Self-Check Status
00 = CRC is in progress.
10 = CRC completed and passed.
01 = CRC completed but failed indicating possible issue with the ROM content. System held in lockup state.
11 = CRC never executed because user disabled CRC check through fuse settings.

| Value | Description |
|---|---|
| 0 | CRC is in progress |
| 2 | CRC completed and passed |


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

| Value | Description |
|---|---|
| 1 | CRC completed but failed indicating possible issue with the ROM content. System held in lockup state. |
| 3 | CRC never executed because user disabled CRC check through fuse settings. |

**Bit 2 – BERR** Bus Error Status

| Value | Description |
|---|---|
| 1 | Bus error incurred while performing the DSU command. |
| 0 | No bus error detected |


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

### 15.10.5. Status B REGISTER

**Name:** STATUSB
**Offset:** 0x108
**Reset:** 0x00000000
**Property:** R

**Table 15-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | HPS | | DBGPRES |
| Access | | | | | | HC/HS/R | | HC/HS/R |
| Reset | | | | | | 0 | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | DCCD1 | DCCD0 | BCCD1 | BCCD0 |
| Access | | | | | HC/HS/R | HC/HS/R | HC/HS/R | HC/HS/R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 10 – HPS** Hot-Plugging Status
1 = Hot-Plugging active
0 = Hot-Plugging inactive

| Value | Description |
|---|---|
| 1 | Hot-Plugging active |
| 0 | Hot-Plugging inactive |

**Bit 8 – DBGPRES** Debugger Present Status
This bit is set when a debugger probe is detected.
Only a POR or external reset can reset this bit.

| Value | Description |
|---|---|
| 1 | Debugger probe detected |
| 0 | No debugger detected |

**Bits 2, 3 – DCCD** Debug Communication Channel x Dirty

| Value | Description |
|---|---|
| 1 | DCCDx.DATA register was written to. |
| 0 | DCCDx.DATA register was read from. |


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

**Bits 0, 1 – BCCD** Boot ROM Communication Channel x Dirty

| Value | Description |
|---|---|
| 1 | BCCDx.DATA register was written to. |
| 0 | BCCDx.DATA register was read from. |


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

### 15.10.6. Boot ROM Channel n REGISTER

**Name:** BCC
**Offset:** 0x0110 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** R/W

**Table 15-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 used for ROM commands/responses. Writing to the register sets STATUCB.BCCDn.
Reading from the register clears STATUSB.BCCDn.
Writing BCCx sets STATUSB.BCCx.
Reading BCCx clears STATUSB.BCCx.

**Notes:**

1. BCC[0] and DCC[0] are aliases that use the same internal 32-bit register.
2. BCC[1] and DCC[1] are aliases that use the same internal 32-bit register.


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

### 15.10.7. Debug Communication Channel n REGISTER

**Name:** DCC
**Offset:** 0x0118 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** R/W

**Table 15-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 used for debug communication. Writing to the register sets STATUCB.DCCDn. Reading from the register clears STATUSB.DCCDn.
Writing DCCx sets STATUSB.DCCx.
Reading DCCx clears STATUSB.DCCx.

**Notes:**

1. BCC[0] and DCC[0] are aliases that use the same internal 32-bit register.
2. BCC[1] and DCC[1] are aliases that use the same internal 32-bit register.


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

### 15.10.8. Device Identification Register

**Name:** DID
**Offset:** 0x120
**Reset:** 0x00000053
**Property:** R/W

**Table 15-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | VER[3:0] | | | | PNMID[7:4] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | PNMID[3:0] | | | | PNDID[7:4] | | | |
| Access | R | R | R | R | P/R/W | P/R/W | P/R/W | P/R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PNDID[3:0] | | | | MANID[10:7] | | | |
| Access | P/R/W | P/R/W | P/R/W | P/R/W | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

**Bits 31:28 – VER[3:0]** Version
Provides the version code where a value of 0 represents A.0

**Bits 27:20 – PNMID[7:0]** Device Part Number
Provides bits [15:8] of the Device Part Number, where the entire Part Number is given by PN[15:0] = DID[27:12].

**Bits 19:12 – PNDID[7:0]** Product Device ID
This bit field identifies a device within a product. Refer to the *Ordering Information* for device configurations and corresponding values for Flash memory density, pin count, and device variant.

**Bits 11:1 – MANID[10:0]** Manual ID
This is duplicate of the JEP-106 CC and ID code present in the Device Service Unit Coresight ROM table that always read 0x29.

**Bit 0 – FV** Fixed Value
Fixed Value of 1


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

### 15.10.9. Debugger Access Level REGISTER

**Name:** DAL
**Offset:** 0x124
**Reset:** 0x00000000
**Property:** R

**Table 15-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | CPU3[1:0] | | CPU2[1:0] | | CPU1[1:0] | | CPU0[1:0] | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 2:3, 4:5, 6:7 – CPU**

| Value | Name | Description |
|---|---|---|
| 2 | FULL_DEBUG | Debugger has full access to debug CPUn |
| 0 | SECURED | Debugger access is disabled |

**Bits 1:0 – CPU0[1:0]**

Writing in this bitfield has no effect.
0x0 = (SECURED) Debugger targeting CPU0 domain can only access the DSU external address space otherwise debugger access is disabled
0x2 = (FULL_DEBUG) Debugger can access secure and non-secure regions.
0x3 = Reserved.

| Value | Name | Description |
|---|---|---|
| 2 | FULL_DEBUG | Debugger can access secure and non-secure regions |
| 0 | SECURED | Debugger targeting CPU0 domain can only access the DSU external address space otherwise debugger access is disabled |


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

### 15.10.10. CoreSight ROM Table Entry N Register

**Name:** ENTRY0
**Offset:** 0x1000
**Reset:** 0x00000002
**Property:** R

**Table 15-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** Entry Present
This bit is set to 0 by hardware under the following conditions DAL.CPU0 == DAL0

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

### 15.10.11. CoreSight ROM Table Entry N Register

**Name:** ENTRY1
**Offset:** 0x1004
**Reset:** 0x00000002
**Property:** R

**Table 15-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** Entry Present
This bit is set to 0 by hardware under the following conditions DAL.CPU0 == DAL0

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

## 15.10.12. CoreSight ROM Table Entry N Register

**Name:** ENTRY2
**Offset:** 0x1008
**Reset:** 0x00000002
**Property:** R

**Table 15-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** Entry Present
This bit is set to 0 by hardware under the following conditions DAL.CPU0 == DAL0 or n > DSU_CPU_NUM, meaning the CPU interface does not exist, or ext_cpu_me[n] == 0 when n <= DSU_CPU_NUM, meaning the CPU exists but is disabled otherwise this bit is set to DSU_CORESIGHT_ENTRY[n][0].

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

### 15.10.13. CoreSight ROM Table Entry N Register

**Name:** ENTRY3
**Offset:** 0x100C
**Reset:** 0x00000002
**Property:** R

**Table 15-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** Entry Present
This bit is set to 0 by hardware under the following conditions DAL.CPU0 == DAL0 or n > DSU_CPU_NUM, meaning the CPU interface does not exist, or ext_cpu_me[n] == 0 when n <= DSU_CPU_NUM, meaning the CPU exists but is disabled otherwise this bit is set to DSU_CORESIGHT_ENTRY[n][0].

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

## 15.10.14. CoreSight ROM Table Entry 4 Register

**Name:** ENTRY4
**Offset:** 0x1010
**Reset:** 0x00000000
**Property:** R

**Table 15-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** CoreSight Entry Present

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

## 15.10.15. CoreSight ROM Table Entry 5 Register

**Name:** ENTRY5
**Offset:** 0x1014
**Reset:** 0x00000000
**Property:** R

**Table 15-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** CoreSight Entry Present

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

### 15.10.16. CoreSight ROM Table Entry 6 Register

**Name:** ENTRY6
**Offset:** 0x1018
**Reset:** 0x00000000
**Property:** R

**Table 15-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** CoreSight Entry Present

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

### 15.10.17. CoreSight ROM Table Entry 7 Register

**Name:** ENTRY7
**Offset:** 0x101C
**Reset:** 0x00000000
**Property:** R

**Table 15-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[19:12] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[11:4] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADDOFF[3:0] | | | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

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

**Bit 1 – FMT** CoreSight Rom Table Format

| Value | Description |
|---|---|
| 1 | 32-bit format |
| 0 | 8-bit format |

**Bit 0 – EPRES** CoreSight Entry Present

| Value | Description |
|---|---|
| 1 | Entry present |
| 0 | Entry not present |


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

### 15.10.18. CoreSight ROM Table Memory Type Register

**Name:** MEMTYPE
**Offset:** 0x1fcc
**Reset:** 0x00000000
**Property:** R

**Table 15-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SYSMEM |
| Access | | | | | | | | HS/HC/R |
| Reset | | | | | | | | 0 |

**Bit 0 – SYSMEM** System Memory

Value of this register field is set to 1 when CPU0 effective DAL is DAL1 or DAL2 otherwise set to 0.
This bit indicates whether system memory is present on the bus that connects to the ROM table.
This bit is set at power-up when DAL.CPU0 is greater than 0 indicating that the system memory is accessible from a debug adapter.
This bit is cleared at power-up when DAL.CPU0 is equal to 0 indicating that the system memory is not accessible from a debug adapter.

| Value | Description |
|---|---|
| 1 | System memory is also present on this bus |
| 0 | System memory not present on bus. This is a dedicated debug bus. |


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

### 15.10.19. CoreSight Peripheral Identification 4 Register

**Name:** PID4
**Offset:** 0x1fd0
**Reset:** 0x00000000
**Property:** R

**Table 15-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | SIZE[3:0] | | | | DES2[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

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

**Bits 3:0 – DES2[3:0]**
These bits will always return zero when read, indicating a Microchip device.


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

### 15.10.20. CoreSight Peripheral Identification 5 Register

**Name:** PID5
**Offset:** 0x1fd4
**Reset:** 0x00000000
**Property:** R

The bits in this register are unimplemented and always read as '0'.

**Table 15-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | RES0[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – RES0[7:0]**
Always returns 0x00.


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

### 15.10.21. CoreSight Peripheral Identification 6 Register

**Name:** PID6
**Offset:** 0x1fd8
**Reset:** 0x00000000
**Property:** R

The bits in this register are unimplemented and always read as '0'.

**Table 15-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | RES0[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – RES0[7:0]**
Always returns 0x00.


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

### 15.10.22. CoreSight Peripheral Identification 7 Register

**Name:** PID7
**Offset:** 0x1fdc
**Reset:** 0x00000000
**Property:** R

The bits in this register are unimplemented and always read as '0'.

**Table 15-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | RES0[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – RES0[7:0]**
Always returns 0x00.


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

### 15.10.23. CoreSight Peripheral Identification 0 Register

**Name:** PID0
**Offset:** 0x1fe0
**Reset:** 0x000000D0
**Property:** R

**Table 15-26.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PART0[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

**Bits 7:0 – PART0[7:0]**
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.

### 15.10.24. CoreSight Peripheral Identification 1 Register

**Name:** PID1
**Offset:** 0x1fe4
**Reset:** 0x0000009C
**Property:** R

**Table 15-27.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | DES0[3:0] | | | | PART1[3:0] | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

**Bits 7:4 – DES0[3:0]**
These bits will always return 0x9 when read, indicating a Microchip device (Microchip JEP-106 identity code is 0x29).

**Bits 3:0 – PART1[3:0]**
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.

### 15.10.25. CoreSight Peripheral Identification 2 Register

**Name:** PID2
**Offset:** 0x1fe8
**Reset:** 0x0000000A
**Property:** R

**Table 15-28.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | REVISION[3:0] | | | | JEDEC | DES1[2:0] | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

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

**Bit 3 – JEDEC** JEDEC Value
Always returns 0x1 which indicates a JEDEC value is used.

**Bits 2:0 – DES1[2:0]**
These bits will always return 0x2 when read, indicating a Microchip device (Microchip JEP-106 identity code is 0x29).


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

### 15.10.26. CoreSight Peripheral Identification 3 Register

**Name:** PID3
**Offset:** 0x1fec
**Reset:** 0x00000000
**Property:** R

**Table 15-29.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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] | | | | CMOD[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]**
These bits will always return 0x0 when read.

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


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

### 15.10.27. CoreSight Component Identification 0 Register

**Name:** CID0
**Offset:** 0x1ff0
**Reset:** 0x0000000D
**Property:** R

**Table 15-30.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PRMBL0[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

**Bits 7:0 – PRMBL0[7:0]**
These bits will always return 0xD when read.


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

## 15.10.28. CoreSight Component Identification 1 Register

**Name:** CID1
**Offset:** 0x1ff4
**Reset:** 0x00000010
**Property:** R

**Table 15-31.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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] | | | | PRMBL1[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]**
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 www.arm.com).

**Bits 3:0 – PRMBL1[3:0]**
These bits will always return 0x00 when read.


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

### 15.10.29. CoreSight Component Identification 2 Register

**Name:** CID2
**Offset:** 0x1ff8
**Reset:** 0x00000005
**Property:** R

**Table 15-32.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PRMBL2[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |

**Bits 7:0 – PRMBL2[7:0]**
These bits will always return 0x5 when read.


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

### 15.10.30. CoreSight Component Identification 3 Register

**Name:** CID3
**Offset:** 0x1ffc
**Reset:** 0x000000B1
**Property:** R

**Table 15-33.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PRMBL3[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

**Bits 7:0 – PRMBL3[7:0]**
These bits will always return 0xB1 when read.


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

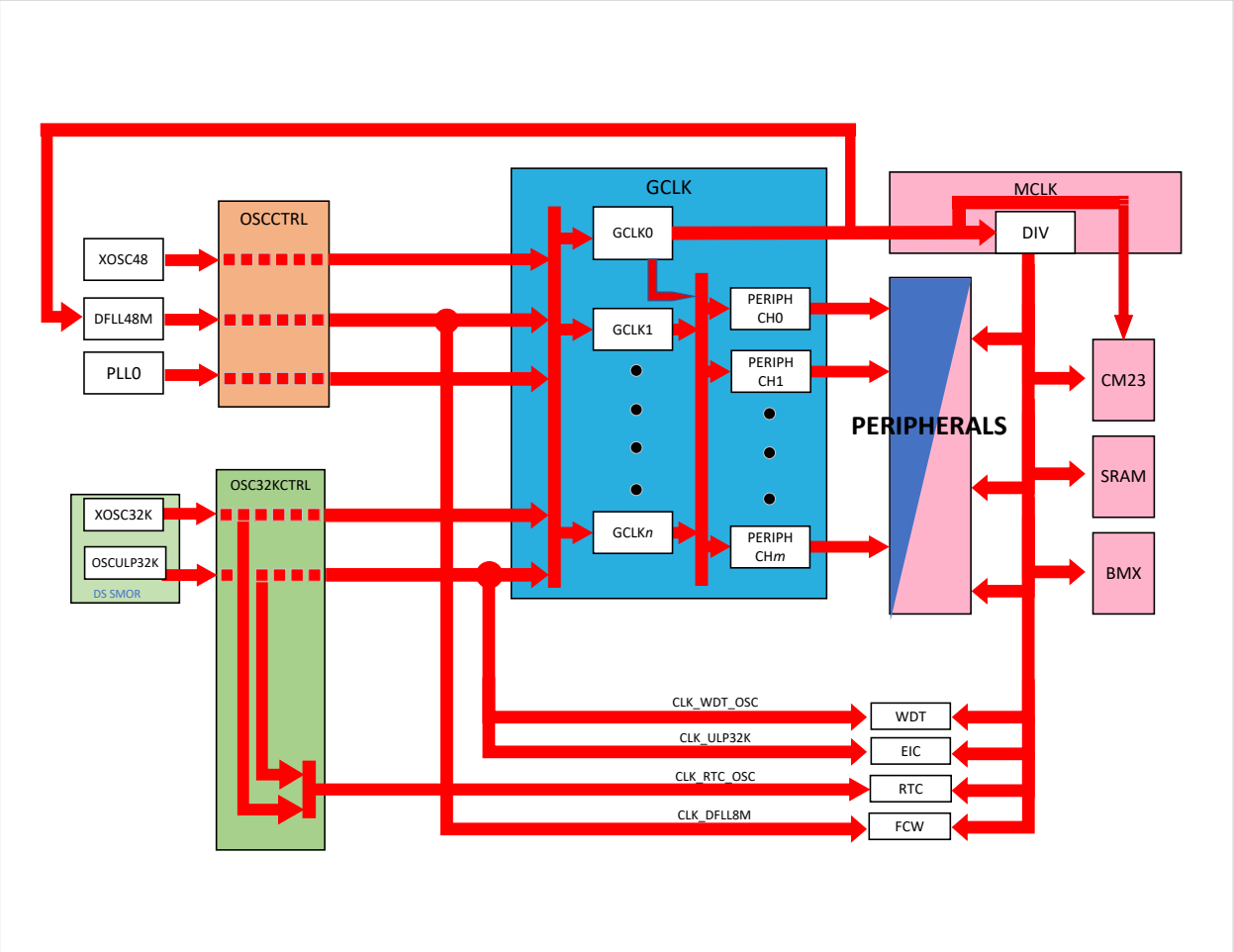
# 16. Clock Distribution System

This chapter summarizes the clock distribution and terminology in the PIC32CM GC00 family of devices. For additional informatin refer to the peripheral chapters and the *GCLK Generic Clock Controller* documentation.

## 16.1. Clock System

### 16.1.1. Overview

**Figure 16-1.** Clock Distribution Block Diagram




The PIC32CM GC00 clock system consist of the following:

- *Clock sources*, that is 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 are the External Crystal Oscillator (XOSC48) and Digital Frequency Locked Loop (DFLL48M).
  - There is 1 PLL available as clock source, for system functions.
- *Generic Clock Controller (GCLK),* which generates, controls, and distributes asynchronous clocks consisting of:
  - *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


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

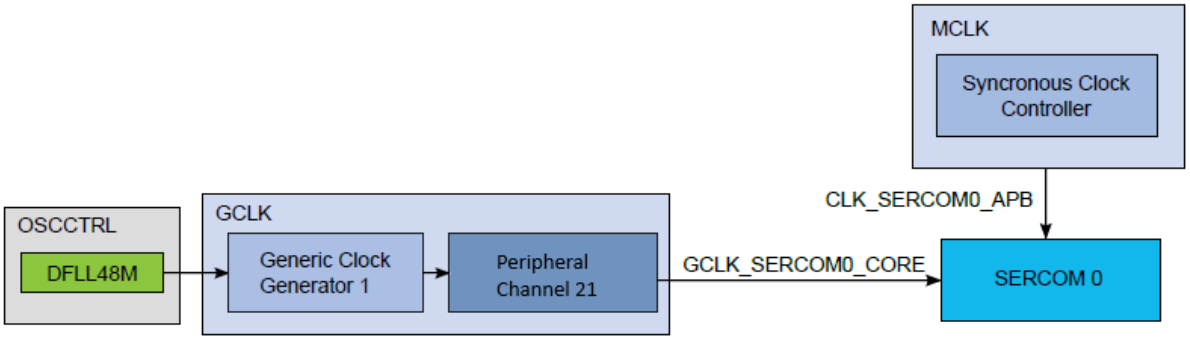
GCLK_- MAIN, which is used by the Power Manager and the Main Clock (MCLK) module, which in turn generates synchronous clocks.

  - *Generic Clocks:* These are clock signals generated by Generic Clock Generators. They are 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 for the system. This includes the CPU, bus clocks (AHB (Advanced High-performance Bus)), APB (Advanced Peripheral Bus) and AXI (Advanced eXtensible Interface)), as well as the Special Function Register 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.

Following figure 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 as GCLK_SERCOM0_CORE, is connected to SERCOM0. The SERCOM0 interface is clocked by CLK_SERCOM0_APB.

**Figure 16-2.** 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, (n = 0-15), 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 the Mapping table in the description of GCLK.PCHCTRLm, (i.e., *GCLK* chapter).
- The AHB clocks are enabled and disabled by writing to the respective bit in the AHB Mask register (MCLK.CLKMASKn).

The APB clocks are enabled and disabled by writing to the respective bit in the APB x Mask registers (MCLK.CLKMASKn).

### 16.1.2. Synchronous and Asynchronous Clocks

As the CPU and the peripherals can be in different clock domains, peripheral accesses by the CPU need 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.

Synchronous Clocks are generated by the CPU and bus clocks Main Clock Controller (MCLK), while asynchronous clocks are generated by the Generic Clock Controller (GCLK).


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

### 16.1.3. 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 as selected in the OSCCTRL Module.
- A clock from the Generic Clock Generator must be configured to use one of the running Clock Sources, and the Generator must be enabled.
- 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.

#### 16.1.3.1. On Demand Clock Requests

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, through the GCLK, to the clock source. If one or more peripheral is using a clock source, the clock source will be started or 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:

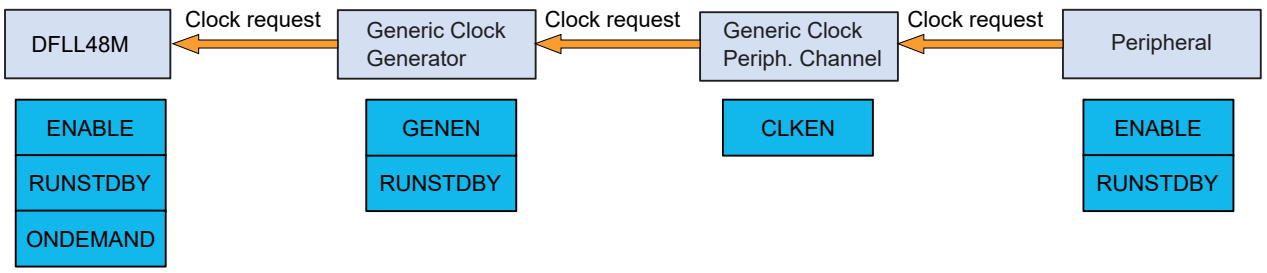
$T_{stop_min}$ = 1 × divided clock source period + 1 × clock source period

$T_{stop_max}$ = 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 as shown in the figure below:

**Figure 16-3.** Clock Request Routing




#### 16.1.3.2. Power Consumption Versus 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: clocking a peripheral with a very low clock, the active power consumption of the peripheral will be lower. At


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

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.

#### 16.1.3.3. Clocks after Reset

On any Reset the synchronous clocks start to their initial state:

- DFLL48M is enabled and configured to run at 48 MHz Open Loop
- Generic Clock Generator 0 uses DFLL48M by default as a source and generates GCLK_MAIN and CLK_MAIN
- CPU and BUS clocks are undivided and enabled

On a Power-on Reset, the 32 kHz clock sources are Reset and the GCLK module starts to its initial state:

- All Generic Clock Generators are disabled except Generator 0
- 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, for example, the according WRTLOCK is set to 1 prior to Reset

#### 16.1.3.4. Clocks During Boot ROM Code Execution

An important point is that the clocks during boot code execution must be limited to 48 MHz not 72 MHz. This is because the device does not know the temperature environment and the clocks are limited to 48 MHz at > 85°C. Therefore, it is recommended that the boot code initialize the DFLL48M to provide 48 MHz source clock and MCLK derive system clocks from the DFLL48M. The PLL is not used.

#### 16.1.3.5. Clocks After Boot ROM Code Execution

Boot code will exit to the BFM application boot code running on the DFLL48M clock.

The application boot code is then responsible for programming the PLL and clock system for the desired system clocks.


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

# 17. Oscillator Controller (OSCCTRL)

## 17.1. Overview

The Oscillators Controller (OSCCTRL) provides a user interface to the XOSC and oscillator used during start-up operation (DFLL). Each oscillator is controlled by dedicated registers in the OSCCTRL. Through this interface, the oscillators are enabled, disabled, or have their configuration 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. Generated events can be used by the system to trigger actions on other peripherals.

## 17.2. Features

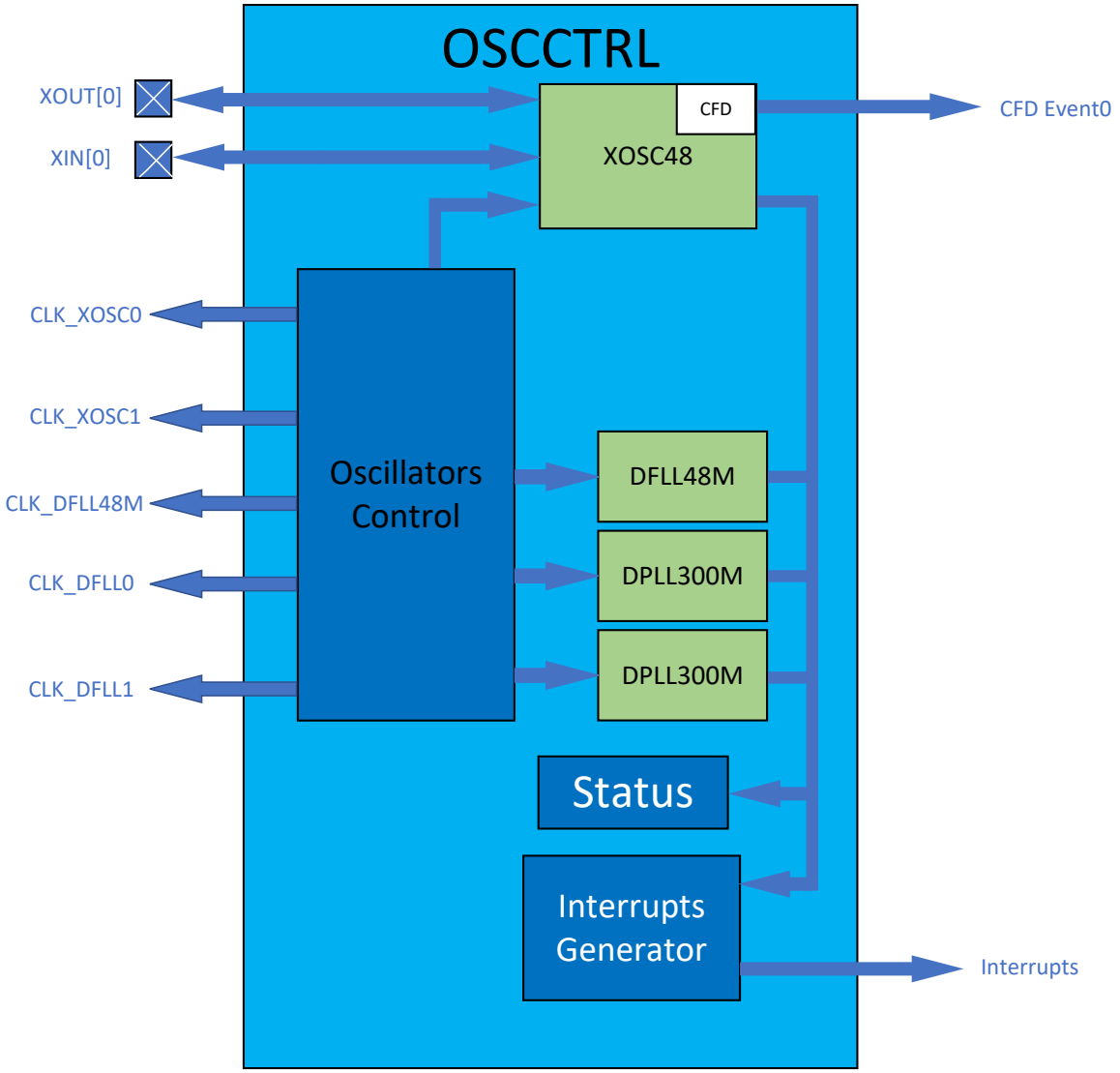
The following are key features of the OSCCTRL module:

- 48 MHz Crystal Oscillator (XOSC48)
  - Supports up to two 4 MHz to 48 MHz crystal oscillators
  - 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
- 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
- Up to 2 Digital Phase Locked Loop (PLLn)
  - Refer to *PLL specifications* for input and output frequency ranges
  - Each DPLL with up to 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 1/16th of reference clock step
  - Embedded test mode controller


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

## 17.3. OSCCTRL Block Diagram

**Figure 17-1.** OSCCTRL Block Diagram




## 17.4. Signal Descriptions

| Signal Name | Type | Description |
|---|---|---|
| $X_{IN}$ | Analog Input | Primary Multipurpose Crystal Oscillator or external clock generator input |
| $X_{OUT}$ | Analog Output | Primary Multipurpose Crystal Oscillator output |


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

## 17.5. Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AXI/APB Clock Index | PAC Peripheral Identifier (PAC.WRCTRL) | Power Domain |
|---|---|---|---|---|---|
| OSCCTRL | 0x4404_0000 | 5 : XOSCFAIL, CLKFAIL<br>6: XOSCRDY<br>7: DFLLRDY<br>8: DFLLLOCK, DFLLLOVF, DFLLUNF, DFLLRCS, DFLLFAIL<br>9: PLLLOCKR0, PLLLOCKR1, | MCLK.CLKMSK0[9] | 6 : INTFLAGA[6] STATUSA[6] | VDDREG |

### 17.5.1. I/O Lines

The XOSC I/O lines are automatically configured when XOSC is enabled. There is no need for user configuration.

### 17.5.2. 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.

### 17.5.3. Clocks

The OSCCTRL gathers controls for all device oscillators and provides clock sources to the Generic Clock Controller (GCLK). The available clock sources are: XOSC, DFLL48M, PLL 1.6GHz, and FRACDIV.

The DFLL48M requires a reference clock (GCLK_DFLL48M_REF) from the GCLK. The control logic uses the oscillator output, which is asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

The PLL requires a reference clock (GCLK_PLL_REF) from the GCLK when the PLL reference selector PLLCTRL.REFSEL is set to GCLK.

The FRACDIV control logic uses the fractional divider output, which is asynchronous to the user interface clock (CLK_OSCCTRL_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains. Refer to Synchronization for further details.

### 17.5.4. Interrupts

The interrupt request line is connected to the Interrupt Controller. Using the OSCCTRL interrupts requires the interrupt controller to be configured first.

### 17.5.5. Events

The events are connected to the Event System. Using the events requires the Event System to be configured first. For additional information, refer to the *Event System (EVSYS)*.

### 17.5.6. 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.


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

### 17.5.7. Register Access Protection

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

- The Interrupt Flag Status and Clear register (INTFLAG)

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

When the CPU is halted in Debug mode, all write protection is automatically disabled. Write protection does not apply for accesses through an external debugger.

### 17.5.8. Analog Connections

The 4-48 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.

## 17.6. Functional Description

### 17.6.1. Principles of Operation

The XOSC, DFLL48M, and PLL0 are configured through the OSCCTRL control registers. Through this interface, the 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.

### 17.6.2. External Multipurpose Crystal Oscillator (XOSC) Operation

The XOSC can operate in the following modes:

- External clock with an external clock signal connected to the XIN pin.
- Crystal oscillator with an external 4-48 MHz crystal.

The XOSC can be used as a clock source for generic clock generators. This is configured by the Generic Clock Controller.

At reset, the XOSC is disabled, and the XIN/XOUT pins can be used as General Purpose I/O (GPIO) pins or by other peripherals in the system. When XOSC is enabled, the operating mode determines the GPIO usage. When in Crystal Oscillator mode, the XIN and XOUT pins are controlled by the OSCCTRL, and GPIO functions are overridden on both pins. When in external clock mode, only the XIN pins will be overridden and controlled by the OSCCTRL, while the XOUT pin can still be used as a GPIO pin.

The XOSC is enabled by writing a '1' to the Enable bit in the External Multipurpose Crystal Oscillator Control register (XOSCCTRLA.ENABLE). To enable XOSC as an External Crystal oscillator, the XTAL Enable bit (XOSCCTRLA.XTALEN) must be written to '1'. If XOSCCTRLA.XTALEN is zero, the external clock input on XIN will be enabled.

If the External Multipurpose Crystal Oscillator Auto Gain Control Loop (XOSCCTRLA.AGC) is '1', the oscillator transconductance value (gm) will be automatically adjusted starting from lowest setting, and in most cases result in low power consumption.

The Gain control loop update rate can be controlled by the XOSCCTRLB user configuration bits.

Alternatively if the External Multipurpose Crystal Oscillator Auto Gain Control Loop (XOSCCTRLA.AGC) is '0', the user can select the oscillator operating condition by setting the transconductance value in the XOSCCTRLB register.

The XOSC will behave differently in different sleep modes, based on the settings of XOSCCTRLA.ONDEMAND, and XOSCCTRLA.ENABLE. XOSCCTRLA.ONDEMAND must be written when


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

XOSCCTRLA.ENABLE = 0. Otherwise, the write of this bit is ignored. If XOSCCTRLA.ENABLE = 0, the XOSC will be always stopped. For XOSCCTRLA.ENABLE = 1, the table below is valid:

**Table 17-1.** XOSC Sleep Behavior

| CPU Mode | ON DEMAND | Sleep Behavior of XOSC and CFD |
|---|---|---|
| Active or Idle | 0 | Always run |
| Active or Idle | 1 | Run if requested by a peripheral |
| Standby | 0 | Always run |
| Standby | 1 | Run if requested by a peripheral |
| Backup | 0 | Always OFF |
| Backup | 1 | Always OFF |

After a hard reset or when waking up from a sleep mode where the XOSC was disabled, the XOSC 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 (XOSCCTRLA.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.XOSCRDY) is set when the external clock or crystal oscillator is stable and ready to be used as a clock source. The INTFLAG.XOSCRDY bit is set on a zero-to-one transition of STATUS.XOSCRDY and an interrupt is generated if the External Multipurpose Crystal Oscillator Ready bit in the Interrupt Enable Set register (INTENSET.XOSCRDY) is set.

If the External Multipurpose Crystal Oscillator Auto Gain Control Loop (XOSCCTRLA.AGC) is set, and the XOSC fail to oscillate after attempting all gain settings, the External Multipurpose Crystal Oscillator Startup Fail bit in the Status register (STATUS.XOSCFAIL) is set. The INTFLAG.XOSCFAIL bit is set on a zero-to-one transition of STATUS.XOSCFAIL and an interrupt is generated if the XOSC Startup Failure bit in the Interrupt Enable Set register (INTENSET.XOSCFAIL) is set.

After the the startup time has elapsed and the External Multipurpose Crystal Oscillator did not fail oscillating, the output clock can also be monitored for failure by using the Clock Fail Detector (CFD). Refer to next section for CFD operations.

### 17.6.3. 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 (XOSC). It detects failing operation of the XOSC clock and allows to switch to a safe clock in case of clock failure. The safe clock is derived from the DFLL48M with a configurable prescaler. The user can also switch from the safe clock to the XOSC clock in case of clock recovery. This allows to configure the safe clock in order to fulfill the operative conditions of the microcontroller. The CFD operation is automatically suspended when the XOSC 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 XOSC clock when the oscillator is disabled (XOSCCTRLA.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 (XOSCCTRLA.CFDEN). After the start or restart of the XOSC, the CFD does not detect failure until the start-up time, as configured by the Oscillator Start-Up Time (XOSCCTRLA.STARTUP) in the External Multipurpose Crystal Oscillator Control register, is elapsed. Once the XOSC Start-Up Time is elapsed, the XOSC clock is constantly monitored.


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

During a period of 4 safe clocks , the CFD watches for a clock activity from the XOSC. There must be one rising and one falling XOSC 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.CLKFAIL) is set. The Clock Failure Interrupt Flag bit in the Interrupt Flag register (INTFLAG.CLKFAIL) is set. If the CLKFAIL bit in the Interrupt Enable Set register (INTENSET.CLKFAIL) 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.CFDEO) is set.

The XOSC clock continues to be monitored after a clock failure. The Clock Failure Status bit in the Status register (STATUS.CLKFAIL) reflects the current XOSC clock activity.

**Clock Switch Back**

When a clock failure is detected, the XOSC clock is replaced by the safe clock in order to maintain an active clock during the XOSC clock failure. The safe clock source can be down scaled with a configurable prescaler to ensure that the safe clock frequency does not exceed the operating conditions selected by the application. When the XOSC clock is switched to the safe clock, the Clock Switch bit (STATUS.XOSCCKSW) in the Status register is set.

When the XOSC clock is switched to the safe clock, the External Multipurpose Crystal Oscillator Ready bit in the Status register (STATUS.XOSCRDY) is set when the safe clock is stable and ready to be used as a clock source.

When the CFD has switched to the safe clock, the XOSC is not disabled. 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 XOSC, the Clock Failure Status bit in the Status register (STATUS.CLKFAIL) is cleared. The application can switch back to the XOSC clock by writing a one to Switch Back bit (XOSCCTRLA.SWBEN) in the External Oscillator Control register. Once the XOSC clock is switched back, the Switch Back bit (XOSCCTRLA.SWBEN) is cleared by the hardware.

**Prescaler:**

The CFD has an internal configurable prescaler (XOSCCTRLA.CFDPRESC) to generate the safe clock from the DFLL48M clock. The prescaler size allows to scale down the DFLL48M clock such that the safe clock is not higher than the XOSC 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 must select a downscale value above 6 (48/8), for example, 8, therefore CFDPRESC = 3.

**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 XOSC and the peripheral clock request. For further details, refer to the Sleep Behavior table. The CFD interrupt can be used to wake up the device from sleep modes.

### 17.6.4. 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 reference clock to get high accuracy on the output clock (CLK_DFLL48M).

The DFLL48M can be used as a source for the generic clock generators.


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

### 17.6.4.1. Basic DFLL48M Operation

#### DFLL48M Operating modes

The DFLL48M will behave differently in different sleep modes, based on the settings of DFLLCTRLA.ONDEMAND and DFLLCTRLA.ENABLE. The DFLLCTRLA.ONDEMAND must be written when DFLLCTRLA.ENABLE = 0 and DFLLSYNC.ENABLE = 0. Otherwise, the write of this bit is ignored. If DFLLCTRLA.ENABLE = 0, the DFLL48M will be always stopped. For DFLLCTRLA.ENABLE = 1, the table below is valid:

**Table 17-2.** DFLL48M Sleep Behavior

| CPU Mode | ON DEMAND | Sleep Behavior of DFLL48M |
|---|---|---|
| Active or Idle | 0 | Always run |
| Active or Idle | 1 | Run if requested by a peripheral |
| Standby | 0 | Always run |
| Standby | 1 | Run if requested by a peripheral |
| Backup | 0 | Always OFF |
| Backup | 1 | Always OFF |

The DFLL48M is used as a clock source for the generic clock generators, as described in the GCLK chapter.

The DFLL48M is factory calibrated for 48 MHz. The frequency calibration is applied at reset.

#### DFLL48M 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 TUNE register (DFLLTUNE.TUNE).

It is possible to change the values of DFLLTUNE.TUNE 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.

#### 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 tuning is automatically adjusted. The DFLL48M must be correctly configured before the closed-loop operation can be enabled. After enabling the DFLL48M, it must be configured in the following way:

1. Enable and select a reference clock (CLK_DFLL48M_REF). CLK_DFLL48M_REF is Generic Clock Channel 0 (DFLL48M_Reference). Refer to GCLK for details.
2. Select the maximum step size allowed in finding the TUNE values by writing the appropriate values to the DFLL Maximum Step bit group (DFLLMUL.STEP) in the DFLL STEP 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.STEP should not be higher than 50% of the maximum value of DFLLTUNE.TUNE.
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. If the target frequency is below the minimum or above the maximum frequency of the DFLL48M, the output frequency will be equal to the DFLL minimum or maximum frequency. Write the DFLLMUL.MUL to restore the TUNE register (DFLLTUNE.TUNE) to its reset value.
4. Start the closed-Loop mode by writing a '1' to the DFLL Loop Enable bit (DFLLCTRLB.LOOPEN) in the DFLL Control register.

The frequency of CLK_DFLL48M (Fclkdfll48m) is given below:


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

**Equation 17-1.** FCLKDFLL48M

$$F_{CLKDFLL48M} = (DFLLMUL.MUL * F_{CLKDFLL48MREF})$$

Where,

$F_{CLKDFLL48MREF}$ is the frequency of the reference clock (CLK_DFLL48M_REF).

The DFLLTUNE register is read-only in closed-loop mode and is controlled by the frequency tuner to meet the user specified frequency.

### DFLL48M Frequency Locking

In the lock of the frequency search in closed-loop mode, the control logic tunes the value in DFLLTUNE.TUNE so that the output frequency is very close to the desired frequency. On lock, the DFLL Locked bit (STATUS.DFLLLOCK) in the status register will be set.

Interrupts are generated by the rising of STATUS.DFLLLOCK if INTENSET.DFLLOCK is 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 will not be met until the Lock state is reached. For additional information on lock times, refer to the *Electrical Characteristics* chapter.

### DFLL48M 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 (DFLLDIFF.DIFF) in the DFLL DIFF register. The relative error on CLK_DFLL48M compared to the target frequency is calculated as follows:

ERROR = (DIFF / MUL)

### DFLL48M 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 DFLLTUNE.TUNE can change after every measurement of CLK_DFLL48M_REF. If the DFLLTUNE.TUNE value overflows or underflows due to large drift in temperature and/or voltage, the DFLL Overflow or Underflow bits (STATUS.DFLLOVF or STATUS.DFLLUNF) in the Status register will be set. After an overflow or underflow 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.DFLLOVF or STATUS.DFLLUNF if the DFLL Overflow or Underflow bit (INTENSET.DFLLOVF or STATUS.DFLLUNF) in the Interrupt Enable Set register is set. If the Stable DFLL Frequency bit (DFLLCTRLB.STABLE) in the DFLL Control register is one, the DFLLTUNE.TUNE 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 (DFLLDIFF.DIFF).

### DFLL48M Reference Clock Stop Detection

If CLK_DFLL48M_REF stops or is running at a very low frequency (slower than CLK_DFLL48M/(2^17)), 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.

### DFLL48M Low Frequency, Low-Power mode

The DFLL48M oscillator can operate in both open loop or closed loop at a reduced frequency of 8 MHz. To select the low frequency or low power, set the Low Frequency bit (LOWFREQ) in the DFLL48M register. This bit is enable protected and can be changed only when the DFLL48M is


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

disabled. In closed-loop the user should adjust the multiplier value DFLLMUL.MUL depending on the frequency of the reference clock (CLK_DFLL48M_REF) to have an output clock frequency of 8 MHz.

#### 17.6.4.2. Additional DFLL48M Features

**Dealing With Delay in the DFLL48M 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 lock. 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 '1' to the DFLL Chill Cycle Disable bit (DFLLCTRLB.CCDIS) in the DFLL Control register. Enabling chill cycles might double the lock time.

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 '1' 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.

**DFLL48M Lose Lock After Wake**

DFLL48M can optionally reset its Lock bit when it is disabled or stopped (not requested). This is configured by the Lose Lock After Wake bit (DFLLCTRLB.LLAW) in the DFLL Control register. If DFLLCTRLB.LLAW is zero, when the DFLL48M is re-enabled or requested again, it starts running with the same configuration as before being disabled, even if the reference clock is not available. The lock will not be lost. Therefore it is important that the user checks that the DFLL48M has reached the lock stage before entering a sleep mode. When the reference clock has restarted, the TUNE tracking will quickly compensate for any frequency drift during sleep if DFLLCTRLB.STABLE is '0'. If DFLLCTRLB.LLAW is '1' when disabling or stopping the DFLL48M, the DFLL48M will lose its lock and needs to regain it through the full lock sequence.

**DFLL48M 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 '0', 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 Lock bit (STATUS.DFLLCK) 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.

**DFLL48M Accuracy**

The following two main factors that determine the accuracy of DFLL48M which can be tuned to obtain maximum accuracy when fine lock is achieved:

- Resolution: The frequency step between two tune values.
- The accuracy of the reference clock.

**DFLL48M Backup Oscillator**

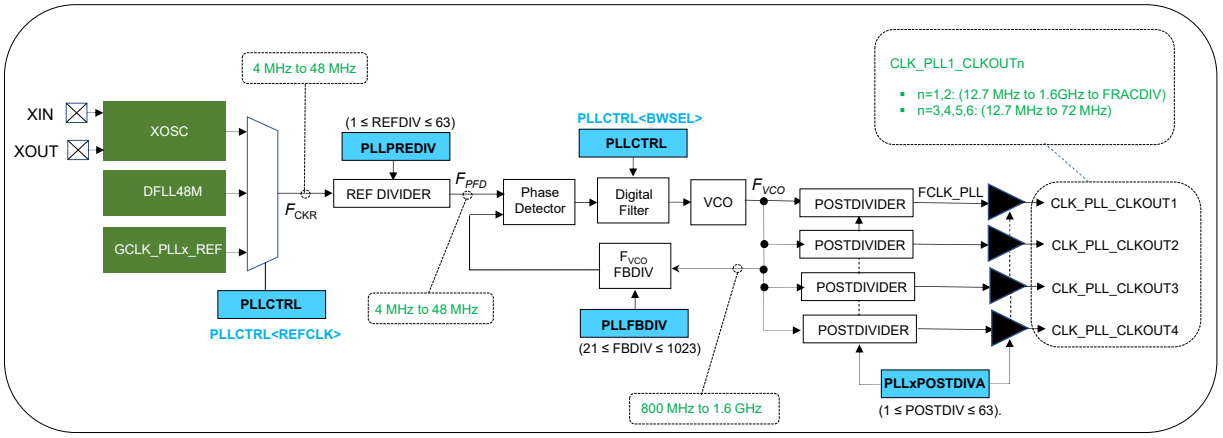
The DFLL48M has an alternate 6 MHz backup oscillator. At any DFLL48M oscillator power-up, the 6 MHz backup oscillator is started and watches for DFLL48M oscillations. If no oscillation is present after the longest DFLL48M startup time, the DFLL48M clock is switched to the 6 MHz backup oscillator.

The DFLL Startup Failure bit in the STATUS register (STATUS.DFLLFAIL) will be set. The INTFLAG.DFLLFAIL bit is set on a zero-to-one transition of STATUS.DFLLFAIL and an interrupt is generated if the DFLL Startup Failure bit in the Interrupt Enable Set register (INTENSET.DFLLFAIL) is set.


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

## 17.6.5. Phase Locked Loop (PLL) Operation

**Figure 17-2.** PLL Block Diagram




### 17.6.5.1. PLL Basic Operation

#### 17.6.5.1.1. PLL Initialization, Enabling, and Disabling

The PLL is enabled by writing a '1' to the Enable bit in the Control register (PLLCTRL.ENABLE). The PLL is disabled by writing a '0' to PLLCTRL.ENABLE.

**Note:** If the PLL is active, the user must ensure that at no time do they violate the minimum or maximum frequency ranges specified for FPFD, FVCO, and FCLK_PLL. Failure to do so destabilize the VCO and lead to unpredictable behavior. Therefore carefully select the order in which the user configures the various scaler values in the PLL, based on the FCKR clock source frequencies used in the application.

#### 17.6.5.1.2. PLL Reference selection

The PLL controller supports three independent sources of reference clock:

- **DFLL48M:** The reference clock is provided by the DFLL48M.
- **XOSC:** The reference clock is provided by the External Multipurpose Crystal Oscillator (XOSC).
- **GCLK:** the reference clock is provided by the Generic Clock Controller.

The reference source can be selected by setting the REFSEL bits in the PLLCTRL register. The frequency of the reference must be comprised between 4 MHz and 60 MHz.

#### 17.6.5.1.3. PLL Settings

The frequency generated by the PLL is determined by the following control registers:

- **PLL REFDIV:** The PLL Reference Frequency Divider, 1 ≤ REFDIV ≤ 63
- **PLL BWSEL:** The PLL Bandwidth Selection, 1 ≤ BWSEL ≤ 4.
- **PLLFBDIV:** The PLL Feed Back Frequency Divider, 21 ≤ FBDIV ≤ 1023
- **POSTDIV:** The PLL Output Frequency Divider, 1 ≤ POSTDIV ≤ 63

When the controller is enabled, the relationship between the reference clock frequency and the output clock frequency is given in the equation below:

FCLK_PLL$F_{CLK_PLL} = (F_{CKR} * (FBDIV / (REFDIV * POSTDIV)))$

Where,

- $F_{CLK_PLL}$ is the frequency of the PLL output clock


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

- $F_{CKR}$ is the frequency of the selected reference clock
- REFDIV is the reference prescaler value
- FBDIV is the loop divider value
- POSTDIV is the output prescaler value

**Example:** (REFDIV = 2,POSTDIV = 5): assuming fCKR = 4 MHz and fCLK_PLL = 200 MHz, the Feed Back divider FBDIV shall be set to 500.

The frequency after the reference divider (FPDF) is given by the formula:

$\mathbf{F_{PFD}} = F_{CKR} / REFDIV$

The frequency of the Voltage Controlled Oscillator (VCO) giving the PLL oscillation is given by the formula:

$\mathbf{F_{VCO}} = F_{CKR} * (FBDIV / REFDIV)$ (must be between 800 MHz and 1600 MHz)

**Note:** $F_{CKR}$ , REFDIV and FBDIV must be selected to satisfy these conditions.

**Notes:**

1. The PLLREFDIV and PLLFBDIV registers are not write-protected by the PAC. Alternatively, they can be write protected by setting the Control bit ( PLLCTRL.WRTLOCK). When the OSCCTRL is PAC write-protected, the user can still tune the PLL frequency when the PLLCTRL.WRTLOCK is cleared, or the user can also write-protect the PLLREFDIV and PLLFBDIV registers when PLLCTRL.WRTLOCK is set.
2. Each PLL has up to four outputs. Each output has an individual PLL output frequency divider POSTDIVn and an individual output control enable OUTENn with n = 0..3. Depending on the selected REFDIV and the reference frequency, the user must set the Band Width Selection bits (BWSEL) in the PLLCTRL register. Refer to the PLLCTRL.BWSEL definition in the “Register Description” section.

#### 17.6.5.1.4. PLL Lock and Clock Generation

After the PLL oscillator is enabled, the PLL controller waits for the oscillator to issue a Lock status. The PLLLOCK bit in the Status register (STATUS.PLLLOCK) will be set and the PLL controller issue the clock to the system (GCLK). The frequency of the PLL output clock CLK_PLL is stable when the STATUS.PLLLOCK bit is set. The PLL Lock Rising bit (PLLLOCKR) in the INTFLAG register is set when the STATUS.PLLLOCK rises.

**Note:** During each PLL startup phase, the clock to the internal modules is not delivered as long as the first lock is not detected. When the lock is detected, the clock is released to the GCLK as long as requested.
The PLL Lock Falling bit (PLLLOCKF) in the INTFLAG register is set when the STATUS.PLLLOCK falls.

The PLL outputs whose OUTEN bit are set will start issuing a clock after the rise of the PLL LOCK Status bit (STATUS.PLLLOCK).

#### 17.6.5.1.5. PLL Disabling

The PLL is disabled by writing a ‘0’ to PLLCTRL.ENABLE. Due to the synchronization of control and configuration registers, generation of internal timings to stop and power-down properly, the PLL will be active for a few microseconds after CTRLA.ENABLE is cleared. The end of the PLL activity can be checked with the Status Lock bit (STATUS.LOCK) being ‘0’. The PLL reference must not be stopped until the STATUS.LOCK is read ‘0’.

#### 17.6.5.1.6. PLL Operation in Sleep Modes

The PLL will behave differently in different sleep modes, based on the settings of PLLCTRL.ONDEMAND and PLLCTRL.ENABLE. PLLCTRL.ONDEMAND must be written when PLLCTRL.ENABLE = 0. Otherwise, the write of this bit is ignored. If PLLCTRL.ENABLE = 0, the PLL will always be stopped. For PLLCTRL.ENABLE = 1, the table below is valid:


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

**Table 17-3.** PLL Sleep Behavior

| CPU Mode | ON DEMAND | Sleep Behavior of DFLL48M |
|---|---|---|
| Active or Idle | 0 | Always run |
| Active or Idle | 1 | Run if requested by a peripheral |
| Standby | 0 | Always run |
| Standby | 1 | Run if requested by a peripheral |
| Backup | 0 | Always OFF |
| Backup | 1 | Always OFF |

#### 17.6.5.1.7. PLL Reference Clock Switching

When a software operation requires reference clock switching, the normal operation is to disable the PLL, modify the PLLCTRL.REFCLK and PLLCTRL.BWSEL to select the desired reference source and activate the PLL again. The CLK_PLL output clock is ready when the STATUS.PLLLOCK bit is set.

#### 17.6.5.1.8. PLL Updates

The PLL Controller supports on-the-fly update of the PLLFBDIV register, therefore it is allowed to modify the feed-back divider when PLL is enabled. At that time the STATUS.PLLLOCK bit is cleared and set again by hardware when the output frequency reached a stable state.

### 17.6.6. OSCCTRL Interrupts

The OSCCTRL has the following interrupt sources:

- **XOSCRDY – Multipurpose Crystal Oscillator Ready:** A “0-to-1” transition on the STATUS.XOSCRDY bit is detected
- **XOSCFAIL – Xosc Startup Failure:** A “0-to-1” transition on the STATUS.XOSCFAIL bit is detected
- **CLKFAIL – Xosc Clock Failure:** A “0-to-1” transition on the STATUS.CLKFAIL bit is detected
- **DFLLRDY – DFLL48m Ready:** A “0-to-1” transition on the STATUS.DFLLRDY bit is detected
- **DFLLLOCK – DFLL48m Lock:** A “0-to-1” transition on the STATUS.DFLLLOCK bit is detected
- **DFLLOVF – DFLL48m Overflow:** A “0-to-1” transition on the STATUS.DFLLOVF bit is detected
- **DFLLUNF – DFLL48m Underflow:** A “0-to-1” transition on the STATUS.DFLLUNF bit is detected
- **DFLLRCS – DFLL48m Reference Clock Stop:** A “0-to-1” transition on the STATUS.DFLLRCS bit is detected
- **DFLLFAIL – DFLL Startup Failure:** A “0-to-1” transition on the STATUS.DFLLFAIL bit is detected
- **PLLLOCK – PLL Lock:** A “0-to-1” transition on the STATUS.PLLLOCK 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 ‘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 is disabled or the OSCCTRL is reset. See the INTFLAG register for details on how to clear interrupt flags.

The OSCCTRL has fewer request lines than 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.

### 17.6.7. OSCCTRL Events

The Clock Fail Detect (CFD) can generate the following output event:


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

- Clock Failure (CLKFAIL): Generated when the XOSC Clock Failure Status bit is set in the Status register (STATUS.CLKFAIL). The CFD event is not generated when the XOSC Clock Switch bit (STATUS.XOSCCKSW) 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 additional information on configuring the event system.

### 17.6.8. OSCCTRL Synchronization

Due to 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:

- The ENABLE bit in the DFLLCTRLA register – write-synchronized
- The DFLLCTRLB register – read-synchronized
- The DFLLTUNE register – read-synchronized and write-synchronized
- The DFLLMUL register – write-synchronized

## 17.7. Peripheral Dependencies

**Table 17-4.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| OSCCTRL | 0x44006000 | 4 : XOSCRDY/ XOSCFAIL/ CLKFAIL<br>5 : DFLLRDY/ DFLLLOCK/ DFLLOVF/ DFLLUNF/ DFLLRCS/ DFLLFAIL<br>6 : PLLLOCKR_0/ PLLLOCKF_0 | **APB:**<br>CLKMSK[0]<br>MASK6 | **DFLL48M:**<br>PCHCTRL[0]<br>**PLL:**<br>PCHCTRL[1] | 6<br>STATUS0<br>PERID6 | | Generators:<br>**XOSCFAIL:**<br>CHANNEL[2] | PD_CORE_SW |


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

## 17.8. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | EVCTRL | 7:0 | | | | | | | | CFDEO |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | INTENCLR | 7:0 | | | | | | CLKFAIL | XOSCFAIL | XOSCRDY |
| | | 15:8 | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | PLL0LOCKR |
| 0x08 | INTENSET | 7:0 | | | | | | CLKFAIL | XOSCFAIL | XOSCRDY |
| | | 15:8 | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | PLL0LOCKR |
| 0x0C | INTFLAG | 7:0 | | | | | | CLKFAIL | XOSCFAIL | XOSCRDY |
| | | 15:8 | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | PLL0LOCKR |
| 0x10 | STATUS | 7:0 | | | | | XOSCCKSW | CLKFAIL | XOSCFAIL | XOSCRDY |
| | | 15:8 | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | PLLxLOCK |
| 0x14 | XOSCCTRLA | 7:0 | ONDEMAND | | SWBEN | CFDEN | XTALEN | AGC | ENABLE | |
| | | 15:8 | | | | | STARTUP[3:0] | | | |
| | | 23:16 | | | | | CFDPRESC[3:0] | | | |
| | | 31:24 | WRTLOCK | | | | | | USBHSDIV[1:0] | |
| 0x18 | XOSCCTRLB | 7:0 | USRCFG[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | WRTLOCK | | | | | | | |
| 0x1C ... 0x2B | Reserved | | | | | | | | | |
| 0x2C | DFLLCTRLA | 7:0 | ONDEMAND | | | | LOWFREQ | WRTLOCK | ENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x30 | DFLLCTRLB | 7:0 | WAITLOCK | | QLDIS | CCDIS | USBCRM | LLAW | STABLE | LOOPEN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | DFLLTUNE | 7:0 | | TUNE[6:0] | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x38 | DFLLDIFF | 7:0 | DIFF[7:0] | | | | | | | |
| | | 15:8 | DIFF[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x3C | DFLLMUL | 7:0 | MUL[7:0] | | | | | | | |
| | | 15:8 | MUL[15:8] | | | | | | | |
| | | 23:16 | | STEP[6:0] | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x40 | PLL0CTRL | 7:0 | ONDEMAND | | | | | WRTLOCK | ENABLE | |
| | | 15:8 | | | BWSEL[2:0] | | | REFSEL[2:0] | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x44 | PLL0FBDIV | 7:0 | FBDIV[7:0] | | | | | | | |
| | | 15:8 | | | | | | | FBDIV[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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x48 | PLL0REFDIV | 7:0 | | | REFDIV[5:0] | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C | PLL0POSTDIVA | 7:0 | OUTEN | | POSTDIV[5:0] | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x50 | PLL0POSTDIVB | 7:0 | OUTEN | | POSTDIV[5:0] | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x54 ... 0x77 | Reserved | | | | | | | | | |
| 0x78 | SYNCBUSY | 7:0 | FRACDIV1 | FRACDIV0 | DFLLMUL | DFLLDIFF | DFLLTUNE | DFLLCTRLB | DFLLENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x7C ... 0x7F | Reserved | | | | | | | | | |
| 0x80 | XOSCCAL | 7:0 | CAL[7:0] | | | | | | | |
| | | 15:8 | CAL[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 17.8.1. Event Control

**Name:** EVCTRL
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** RW

**Table 17-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CFDEO |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – CFDEO** Clock Failure Detector Event Output Enable
This bit indicates whether the XOSC Clock Failure detector event output is enabled and an output event will be generated when the XOSC Clock Failure detector detects a clock failure.
**Note:** To prevent false event generation, the CFDEO bit must be set or cleared only when the XOSC is disabled (XOSCCTRLn.ENABLE = 0).

| Value | Description |
|---|---|
| 0 | Clock Failure detector event output is disabled and an event will not be generated on a clock fail. |
| 1 | Clock Failure detector event output is enabled and an event will be generated on a clock fail. |


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

### 17.8.2. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** RW

**Table 17-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PLL0LOCKR |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CLKFAIL | XOSCFAIL | XOSCRDY |
| Access | | | | | | RW | RW | RW |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 24 – PLL0LOCKR** PLL 0 Lock Rise Interrupt Enable
Writing a zero to this bit has no effect. Writing a '1' to this bit will clear the PLL Lock Rise Interrupt Enable bit, which disables the PLL Lock Rise interrupt.

| Value | Description |
|---|---|
| 0 | The PLL Lock Rise interrupt is disabled. |
| 1 | The PLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the PLL Lock Rise Interrupt flag is set. |

**Bit 13 – DFLLFAIL** DFLL Startup Failure Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Startup Failure Interrupt Enable bit, which disables the DFLL Startup Failure interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL48M Startup Failure interrupt is disabled. |
| 1 | The DFLL48M Startup Failure interrupt is enabled, and an interrupt request will be generated when the DFLL Startup Failure Interrupt flag is set. |

**Bit 12 – DFLLRCS** DFLL Reference Clock Stopped Interrupt Enable
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 will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL48M Reference Clock Stopped interrupt is disabled. |
| 1 | The DFLL48M Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set. |

**Bit 11 – DFLLUNF** DFLL Tuner Underflow Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Underflow Interrupt Enable bit, which disables the DFLL Tuner Underflow interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL Tuner Underflow interrupt is disabled. |
| 1 | The DFLL Tuner Underflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Underflow Interrupt flag is set. |

**Bit 10 – DFLLOVF** DFLL Tuner Overflow Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Overflow Interrupt Enable bit, which disables the DFLL Tuner Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL Tuner Overflow interrupt is disabled. |
| 1 | The DFLL Tuner Overflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Overflow Interrupt flag is set. |

**Bit 9 – DFLLLOCK** DFLL Lock Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Lock Interrupt Enable bit, which disables the DFLL Lock interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL Lock interrupt is disabled. |
| 1 | The DFLL Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set. |

**Bit 8 – DFLLRDY** DFLL Ready Interrupt Enable
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.

| Value | Description |
|---|---|
| 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. |

**Bit 2 – CLKFAIL** XOSC Clock Failure Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Clock Failure Interrupt Enable bit, which disables the XOSC Clock Failure interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC Clock Failure interrupt is disabled. |
| 1 | The XOSC Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set. |


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

**Bit 1 – XOSCFAIL** XOSC Startup Failure Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Startup Failure Interrupt Enable bit, which disables the XOSC Startup Failure interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC Startup Failure interrupt is disabled. |
| 1 | The XOSC Startup Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Startup Failure Interrupt flag is set. |

**Bit 0 – XOSCRDY** XOSC Ready Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC Ready interrupt is disabled. |
| 1 | The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set. |


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

### 17.8.3. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** RW

**Table 17-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PLL0LOCKR |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CLKFAIL | XOSCFAIL | XOSCRDY |
| Access | | | | | | RW | RW | RW |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 24 – PLL0LOCKR** PLL 0 Lock Rise Interrupt Enable
Writing a zero to this bit has no effect. Writing a '1' to this bit will set the PLL Lock Rise Interrupt Enable bit, which enables the PLL Lock Rise interrupt.

| Value | Description |
|---|---|
| 0 | The PLL Lock Rise interrupt is disabled |
| 1 | The PLL Lock Rise interrupt is enabled, and an interrupt request will be generated when the PLL Lock Rise Interrupt flag is set |

**Bit 13 – DFLLFAIL** DFLL Startup Failure Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Startup Failure Interrupt Enable bit, which disables the DFLL Startup Failure interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL48M Startup Failure interrupt is disabled. |
| 1 | The DFLL48M Startup Failure interrupt is enabled, and an interrupt request will be generated when the DFLL Startup Failure Interrupt flag is set. |

**Bit 12 – DFLLRCS** DFLL Reference Clock Stopped Interrupt Enable
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 will clear the DFLL Reference Clock Stopped Interrupt Enable bit, which disables the DFLL Reference Clock Stopped interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL48M Reference Clock Stopped interrupt is disabled. |
| 1 | The DFLL48M Reference Clock Stopped interrupt is enabled, and an interrupt request will be generated when the DFLL Reference Clock Stopped Interrupt flag is set. |

**Bit 11 – DFLLUNF** DFLL Tuner Underflow Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Underflow Interrupt Enable bit, which disables the DFLL Tuner Underflow interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL Tuner Underflow interrupt is disabled. |
| 1 | The DFLL Tuner Underflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Underflow Interrupt flag is set. |

**Bit 10 – DFLLOVF** DFLL Tuner Overflow Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Tuner Overflow Interrupt Enable bit, which disables the DFLL Tuner Overflow interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL Tuner Overflow interrupt is disabled. |
| 1 | The DFLL Tuner Overflow interrupt is enabled, and an interrupt request will be generated when the DFLL Tuner Overflow Interrupt flag is set. |

**Bit 9 – DFLLLOCK** DFLL Lock Interrupt Enable
Writing a zero to this bit has no effect.
Writing a '1' to this bit will clear the DFLL Lock Interrupt Enable bit, which disables the DFLL Lock interrupt.

| Value | Description |
|---|---|
| 0 | The DFLL Lock interrupt is disabled. |
| 1 | The DFLL Lock interrupt is enabled, and an interrupt request will be generated when the DFLL Lock Interrupt flag is set. |

**Bit 8 – DFLLRDY** DFLL Ready Interrupt Enable
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.

| Value | Description |
|---|---|
| 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. |

**Bit 2 – CLKFAIL** XOSC Clock Failure Interrupt Enable
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Clock Failure Interrupt Enable bit, which disables the XOSC Clock Failure interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC Clock Failure interrupt is disabled. |
| 1 | The XOSC Clock Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Clock Failure Interrupt flag is set. |


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

**Bit 1 – XOSCFAIL** XOSC Startup Failure Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Startup Failure Interrupt Enable bit, which disables the XOSC Startup Failure interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC Startup Failure interrupt is disabled. |
| 1 | The XOSC Startup Failure interrupt is enabled, and an interrupt request will be generated when the XOSC Startup Failure Interrupt flag is set. |

**Bit 0 – XOSCRDY** XOSC Ready Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the XOSC Ready Interrupt Enable bit, which disables the XOSC Ready interrupt.

| Value | Description |
|---|---|
| 0 | The XOSC Ready interrupt is disabled. |
| 1 | The XOSC Ready interrupt is enabled, and an interrupt request will be generated when the XOSC Ready Interrupt flag is set. |


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

### 17.8.4. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** RW

**Note:** Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

**Table 17-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PLL0LOCKR |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CLKFAIL | XOSCFAIL | XOSCRDY |
| Access | | | | | | RW | RW | RW |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 24 – PLL0LOCKR** PLL 0 Lock Rise
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the PLL Lock bit in the Status register (STATUS.PLLLOCK) and will generate an interrupt request if INTENSET.PLLLOCKR is '1'.
Writing a zero to this bit has no effect.
Writing a '1' to this bit clears the PLL Lock Rise interrupt flag.

**Bit 13 – DFLLFAIL** DFLL Startup Failure
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Startup Failure bit in the Status register (STATUS.DFLLFAIL) and will generate an interrupt request if INTENSET.DFLLFAIL is '1'.
Writing a ‘0’ to this bit has no effect.
Writing a '1' to this bit clears the DFLL Startup Failure 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'.


© 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 DFLL Reference Clock Stopped interrupt flag.

**Bit 11 – DFLLUNF** DFLL Tuner Underflow
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Tuner Underflow bit in the Status register (STATUS.DFLLUNF) and will generate an interrupt request if INTENSET.DFLLUNF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DFLL Tuner Underflow interrupt flag.

**Bit 10 – DFLLOVF** DFLL Tuner Overflow
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Tuner Overflow bit in the Status register (STATUS.DFLLOVF) and will generate an interrupt request if INTENSET.DFLLOVF is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DFLL Tuner Overflow interrupt flag.

**Bit 9 – DFLLLOCK** DFLL Lock
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the DFLL Lock bit in the Status register (STATUS.DFLLLOCK) and will generate an interrupt request if INTENSET.DFLLLOCK is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the DFLL Lock 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 '0' to this bit has no effect.
Writing a '1' to this bit clears the DFLL Ready interrupt flag.

**Bit 2 – CLKFAIL** XOSC Clock Failure
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC Clock Failure bit in the Status register (STATUS.CLKFAIL) and will generate an interrupt request if INTENSET.CLKFAIL is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the XOSC Clock Failure interrupt flag.

**Bit 1 – XOSCFAIL** XOSC Startup Failure
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC Startup Failure bit in the Status register (STATUS.XOSCFAIL) and will generate an interrupt request if INTENSET.XOSCFAIL is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the XOSC Startup Failure interrupt flag.

**Bit 0 – XOSCRDY** XOSC Ready
This flag is cleared by writing a '1' to it.
This flag is set on a zero-to-one transition of the XOSC Ready bit in the Status register (STATUS.XOSCRDY) and will generate an interrupt request if INTENSET.XOSCRDY is '1'.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the XOSC Ready interrupt flag.


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

### 17.8.5. Status

**Name:** STATUS
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** R

**Table 17-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PLLxLOCK |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | DFLLFAIL | DFLLRCS | DFLLUNF | DFLLOVF | DFLLLOCK | DFLLRDY |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | XOSCCKSW | CLKFAIL | XOSCFAIL | XOSCRDY |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 24 – PLLxLOCK** PLL x Lock

**Bit 13 – DFLLFAIL** DFLL Startup Failure

| Value | Description |
|---|---|
| 0 | DFLL Startup failure is not detected. |
| 1 | DFLL Startup failure is detected. |

**Bit 12 – DFLLRCS** DFLL Reference Clock Stopped

| Value | Description |
|---|---|
| 0 | DFLL reference clock is running. |
| 1 | DFLL reference clock has stopped. |

**Bit 11 – DFLLUNF** DFLL Tuner Underflow

| Value | Description |
|---|---|
| 0 | DFLL Tuner Underflow is not detected. |
| 1 | DFLL Tuner Underflow is detected. |

**Bit 10 – DFLLOVF** DFLL Tuner Overflow


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

| Value | Description |
|---|---|
| 0 | DFLL Tuner Overflow is not detected. |
| 1 | DFLL Tuner Overflow is detected. |

**Bit 9 – DFLLLOCK** DFLL Lock

| Value | Description |
|---|---|
| 0 | DFLL lock is not detected. |
| 1 | DFLL lock is detected. |

**Bit 8 – DFLLRDY** DFLL Ready

| Value | Description |
|---|---|
| 0 | DFLL is not ready. |
| 1 | DFLL is stable and ready to be used as a clock source. |

**Bit 3 – XOSCCKSW** XOSC Clock Switch

| Value | Description |
|---|---|
| 0 | XOSC is not switched and provides the external clock or crystal oscillator clock. |
| 1 | XOSC is switched and provides the safe clock. |

**Bit 2 – CLKFAIL** XOSC Clock Failure

| Value | Description |
|---|---|
| 0 | XOSC Clock failure is not detected. |
| 1 | XOSC Clock failure is detected. |

**Bit 1 – XOSCFAIL** XOSC Startup Failure

| Value | Description |
|---|---|
| 0 | XOSC Startup failure is not detected. |
| 1 | XOSC Startup failure is detected. |

**Bit 0 – XOSCRDY** XOSC Ready

**Note:** If the CFD is enabled and the XOSC clock is failing, the XOSCRDY status bit remains high, if already set.

| Value | Description |
|---|---|
| 0 | XOSC is not ready. |
| 1 | XOSC is stable and ready to be used as a clock source. |


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

### 17.8.6. External Multipurpose Crystal Oscillator Control A

**Name:** XOSCCTRLA
**Offset:** 0x14
**Reset:** 0x00000D00
**Property:** RW

**Table 17-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | WRTLOCK | | | | | | USBHSDIV[1:0] | |
| Access | RW | | | | | | RW | RW |
| Reset | 0 | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CFDPRESC[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | STARTUP[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 1 | 1 | 0 | 1 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | | SWBEN | CFDEN | XTALEN | AGC | ENABLE | |
| Access | RW | | RW | RW | RW | RW | RW | |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | |

**Bit 31 – WRTLOCK** Write Lock for CTRLA register
**Note:** Once the WRTLOCK bit is set, it can only be cleared by a reset.

| Value | Description |
|---|---|
| 0 | The XOSCCTRLA register can be modified by a system write. |
| 1 | The XOSCCTRLA (except XOSCCTRLA.SWBEN) register is write protected. |

**Bits 25:24 – USBHSDIV[1:0]** USBHS Referrence Clock Division

| Value | Name | Description |
|---|---|---|
| 0 | DIS | USBHS PLL reference XOSC clock is disabled |
| 1 | DIV1 | USBHS PLL reference XOSC clock is divided by 1 |
| 2 | DIV2 | USBHS PLL reference XOSC clock is divided by 2 |
| 3 | DIV4 | USBHS PLL reference XOSC clock is divided by 4 |

**Bits 19:16 – CFDPRESC[3:0]** Clock Failure Detector Prescaler
These bits select the DFLL48oscillator post scaler for the clock fail detector. The CFD safe clock frequency is the DFLL48 frequency divided by 2^CFDPRESC. These bits are XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE=1

| Value | Name | Description |
|---|---|---|
| 0 | DIV1 | 48 MHz |


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

| Value | Name | Description |
|---|---|---|
| 1 | DIV2 | 24 MHz |
| 2 | DIV4 | 12 MHz |
| 3 | DIV8 | 6 MHz |
| 4 | DIV16 | 3 MHz |
| 5 | DIV32 | 1.5 MHz |
| 6 | DIV64 | 0.75 MHz |
| 7 | DIV128 | 0.375 MHz |
| 8 | DIV256 | 187.5 KHz |
| 9 | DIV512 | 93.75 KHz |
| 10 | DIV1024 | 46.875 KHz |
| 11 | DIV2048 | 23.437 KHz |
| 12 | DIV4096 | 11.718 KHz |
| 13 | DIV8192 | 5.85 KHz |
| 14 | DIV16384 | 2.92 KHz |
| 15 | DIV32768 | 1.46 KHz |

**Bits 11:8 – STARTUP[3:0]** Start-Up Time

These bits select start-up time for the oscillator XOSC according to the table below before a clock fail is acknowledged. The OSCULP32K oscillator is used to clock the start-up counter. These bits are XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.

| STARTUP[3:0] | Number of OSCULP32KClock Cycles | Approximate Equivalent Time |
|---|---|---|
| 0x0 | 1 | 31µs |
| 0x1 | 2 | 61µs |
| 0x2 | 4 | 122µs |
| 0x3 | 8 | 244µs |
| 0x4 | 16 | 488µs |
| 0x5 | 32 | 977µs |
| 0x6 | 64 | 1953µs |
| 0x7 | 128 | 3906µs |
| 0x8 | 256 | 7813µs |
| 0x9 | 512 | 15625µs |
| 0xA | 1024 | 31250µs |
| 0xB | 2048 | 62500µs |
| 0xC | 4096 | 125000µs |
| 0xD (Default) | 8192 | 250000µs |
| 0xE | 16384 | 500000µs |
| 0xF | 32768 | 1000000µs |

**Notes:**

1. It is critical when using AGC to allow ample startup time to avoid spurious CFD events.
2. When using AGC BW = 0x0, the minimum startup time is 6.25 ms.

| Value | Name | Description |
|---|---|---|
| 0 | CYCLE1 | 31 us |
| 1 | CYCLE2 | 61 us |
| 2 | CYCLE4 | 122 us |
| 3 | CYCLE8 | 244 us |
| 4 | CYCLE16 | 488 us |
| 5 | CYCLE32 | 977 us |
| 6 | CYCLE64 | 1953 us |
| 7 | CYCLE128 | 3906 us |
| 8 | CYCLE256 | 7813 us |
| 9 | CYCLE512 | 15625 us |


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

| Value | Name | Description |
|---|---|---|
| 10 | CYCLE1024 | 31250 us |
| 11 | CYCLE2048 | 62500 us |
| 12 | CYCLE4096 | 125000 us |
| 13 | CYCLE8192 | 250000 us |
| 14 | CYCLE16384 | 500000 us |
| 15 | CYCLE32768 | 1000000 us |

**Bit 7 – ONDEMAND** On Demand Control

The ONDEMAND operation mode allows the XOSC to be enabled or disabled depending on peripheral clock requests.

**Note:** The XOSC is not running if no peripheral is requesting the clock source.

If ONDEMAND is set, the XOSC will only be running when requested by a peripheral and enabled (XOSCTRLA.ENABLE = 1). If there is no peripheral requesting the XOSC's clock source, the XOSC will be in a disabled state. If ONDEMAND is disabled, the XOSC will always be running when enabled (XOSCTRLA.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active. This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.

| Value | Description |
|---|---|
| 0 | The XOSC is always on. |
| 1 | The XOSC is running when a peripheral is requesting the XOSC to be used as a clock source. |

**Bit 5 – SWBEN** Xosc Clock Switch Back Enable

This bit controls the XOSC output clock switch back to the external clock or crystal oscillator in case of clock recovery.

**Note:** The SWBEN bit is also cleared when the Clock Failure Detector is disabled (CFDEN = 0).

| Value | Description |
|---|---|
| 0 | The clock switch back is disabled. |
| 1 | The clock switch back is enabled. This bit is reset once the XOSC output clock is switched back to the external clock or crystal oscillator. |

**Bit 4 – CFDEN** Clock Failure Detector Enable

This bit controls the XOSC clock failure detector and is enable protected

**Note:** After setting CFDEN to enable clock failure detection, STATUS.CLKFAIL will always be set. This first detection must be ignored. Subsequent setting of this bit will indicate actual clock failure events.

| Value | Description |
|---|---|
| 0 | Clock Failure Detector is disabled. |
| 1 | Clock Failure Detector is enabled. |

**Bit 3 – XTALEN** Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator XOSC.

**Notes:**

1. If XOSCCTRLA.XTALEN = 0 then XOSCCTRLA.AGC = 1 is not permitted.
2. This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.
3. If XOSCCTRLA.XTALEN = 0 then XOSCCTRLB.GMAN (User Manual Gain control) bits are ignored.

| Value | Description |
|---|---|
| 0 | External clock oscillator connected on XIN. XOUT can be used as general-purpose I/O. |
| 1 | Crystal connected to XIN and XOUT. |


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

**Bit 2 – AGC** Auto Gain Control Loop Enable

**Notes:**

- If XOSCCTRLA.XTALEN = 0 then XOSCCTRLA.AGC = 1 is not permitted.
- This bit is XOSCCTRLA.ENABLE protected and cannot be updated if XOSCCTRLA.ENABLE = 1.
- If AGC is enabled, XOSCCTRLB.GMAN (User Manual Gain control) bits are ignored.
- When the XOSCCTRLA.AGC = 1, the Primary Oscillator will automatically do a linear search to find the lowest power/gain setting to guarantee stable oscillation with the user's crystal.

| Value | Description |
|---|---|
| 0 | The oscillator auto gain control loop is disabled. |
| 1 | The oscillator auto gain control loop is enabled. |

**Bit 1 – ENABLE** Oscillator Enable

| Value | Description |
|---|---|
| 0 | The oscillator XOSC is disabled. |
| 1 | The oscillator XOSC is enabled. |


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

### 17.8.7. External Multipurpose Crystal Oscillator Control B

**Name:** XOSCCTRLB
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** RW

**Table 17-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | WRTLOCK | | | | | | | |
| Access | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | USRCFG[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – WRTLOCK** Write Lock for CTRLB register
**Note:** Once the WRTLOCK bit is set, it can only be cleared by a reset.

| Value | Description |
|---|---|
| 0 | The XOSCCTRLB register can be modified by a system write. |
| 1 | The XOSCCTRLB register is write protected. |

**Bits 7:0 – USRCFG[7:0]** User Configuration Control Bits


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

### 17.8.8. DFLL48M Control A

**Name:** DFLLCTRLA
**Offset:** 0x2C
**Reset:** 0x00000082
**Property:** RW

**Table 17-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | | | | LOWFREQ | WRTLOCK | ENABLE | |
| Access | RW | | | | RW | RW | RW | |
| Reset | 1 | | | | 0 | 0 | 1 | |

**Bit 7 – ONDEMAND** On Demand Control

The ONDEMAND operation mode allows the DFLL to be enabled or disabled depending on peripheral clock requests.

**Note:** If ONDEMAND 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 ONDEMAND is disabled the DFLL will always be running when enabled (DFLLTRLA.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active This bit is DFLLCTRLA.ENABLE protected and cannot be updated if DFLLCTRLA.ENABLE = 1.

| Value | Description |
|---|---|
| 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 3 – LOWFREQ** Low Frequency Mode

**Note:** This bit is DFLLCTRLA.ENABLE protected and cannot be updated if DFLLCTRLA.ENABLE = 1.

| Value | Description |
|---|---|
| 0 | The DFLL48M oscillator operates at high frequency. |
| 1 | The DFLL48M oscillator operates at low frequency. |


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

**Bit 2 – WRTLOCK** Write Lock

**Note:** Once the WRTLOCK bit is set, it can only be cleared by a reset.

| Value | Description |
|---|---|
| 0 | The DFLLCTRLA and DFLLCTRLB registers can be modified by a system write. |
| 1 | The DFLLCTRLA and DFLLCTRLB registers are write protected. |

**Bit 1 – ENABLE** DFLL Enable

**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.

| Value | Description |
|---|---|
| 0 | The DFLL48M oscillator is disabled. |
| 1 | The DFLL48M oscillator is enabled. |


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

### 17.8.9. DFLL48M Control B

**Name:** DFLLCTRLB
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** RW

**Note:** In the period of 80-100 us between lock and stabilization, DFLL accuracy will be limited to +/-1.5%. After stabilization has been achieved, the accuracy will be +/-0.25%. Disabling Quick Lock will eliminate this period of inaccuracy.

**Table 17-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | WAITLOCK | | QLDIS | CCDIS | USBCRM | LLAW | STABLE | LOOPEN |
| Access | RW | | RW | RW | RW | RW | RW | RW |
| Reset | 0 | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – WAITLOCK** Wait Lock
This bit controls the DFLL48M output clock, depending on lock status:

| Value | Description |
|---|---|
| 0 | Output clock before the DFLL is locked. |
| 1 | Output clock when DFLL is locked. |

**Bit 5 – QLDIS** Quick Lock Disable

| Value | Description |
|---|---|
| 0 | Quick Lock is enabled. |
| 1 | Quick Lock is disabled. |

**Bit 4 – CCDIS** Chill Cycle Disable

| Value | Description |
|---|---|
| 0 | Chill Cycle is enabled. |
| 1 | Chill Cycle is disabled. |


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

**Bit 3 – USBCRM** USB Clock Recovery Mode

| Value | Description |
|---|---|
| 0 | USB Clock Recovery Mode is disabled. |
| 1 | USB Clock Recovery Mode is enabled. |

**Bit 2 – LLAW** Lose Lock After Wake

| Value | Description |
|---|---|
| 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

| Value | Description |
|---|---|
| 0 | Tune register tracks changes in output frequency. |
| 1 | Tune calibration register value will be fixed after a lock. |

**Bit 0 – LOOPEN** Operating Mode Selection

| Value | Description |
|---|---|
| 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.

### 17.8.10. DFLL48M Tune

**Name:** DFLLTUNE
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** RW

**Table 17-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | TUNE[6:0] | | | | | | |
| Access | | RW | RW | RW | RW | RW | RW | RW |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 6:0 – TUNE[6:0]** Tune Value
Sets the value of the Tune Calibration register.
**Note:** In closed-loop mode, this field is read-only.

| Step | TUNE[6:0] | % Delta/step |
|---|---|---|
| +63 | 0b011 1111 | +9.45% |
| ... | ... | ... |
| +1 | 0b000 0001 | +0.15% |
| 0 | 0b000 0000 / 0b111 1111 | 0% |
| -1 | 0b111 1110 | -0.15% |
| ... | ... | ... |
| -63 | 0b100 0000 | -9.45% |

**Note:**

1. % Delta value is rounded to two decimal places.


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

### 17.8.11. DFLL48M Diff

**Name:** DFLLDIFF
**Offset:** 0x38
**Reset:** 0x00000000
**Property:** R

**Table 17-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | DIFF[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 |
|---|---|---|---|---|---|---|---|---|
| | DIFF[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – DIFF[15:0]** Multiplication Ratio Difference
In closed-loop mode (DFLLCTRLB.LOOPEN is written to one), this bit group indicates the difference between the ideal number of DFLL48M cycles and the counted number of cycles. This value is not updated in open loop mode and should be considered invalid in that case.


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

### 17.8.12. DFLL48M Multiplier

**Name:** DFLLMUL
**Offset:** 0x3C
**Reset:** 0x00000000
**Property:** RW

**Table 17-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | STEP[6:0] | | | | | | |
| Access | | RW | RW | RW | RW | RW | RW | RW |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MUL[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 |
|---|---|---|---|---|---|---|---|---|
| | MUL[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 22:16 – STEP[6:0]** Tune Maximum Step

This bit group indicates the maximum step size allowed during tune adjustment in closed-loop mode. When adjusting to a new frequency, the expected output frequency overshoot depends on this step size.

**Note:** STEP[6:0] is nothing more than the first tuner adjustment value from where the tune search will start. Depending if the current DFLL frequency is slower of faster than the targeted frequency, the next tune register value is *tune+step* or *tune-step*. Each time the current frequency changes from slower to faster or faster to slower, step is divided by 2 until the minimum value of 1.

**Example 1:**
If STEP[6:0] = 64, dichotomic search will be:

- step1 = +/-64
- step2 = +/-32
- step3 = +/-16
- step4 = +/-8
- step5 = +/-4
- step6 = +/-2
- step7 = +/-1


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

**Example 2:**
If STEP[6:0] = 15, dichotomic search will be:

- step1 = +/-15
- step2 = +/-7
- step3 = +/-3
- step4 = +/-1

**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 the lock to be lost and the DFLLTUNE.TUNE register value to be reset to its midpoint, 0b000_0000.
**Example:** CLK_DFLL_REF = XTAL, 32.768 kHz, 100 ppm, DFLLMUL.MUL[15:0] = 1464 = 0x5B8, then DFLL48 = 47.97 MHz.


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

### 17.8.13. PLL Control

**Name:** PLL0CTRL
**Offset:** 0x40
**Reset:** 0x00000000
**Property:** RW

**Table 17-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | BWSEL[2:0] | | | REFSEL[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 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | | | | | WRTLOCK | ENABLE | |
| Access | RW | | | | | RW | RW | |
| Reset | 0 | | | | | 0 | 0 | |

**Bits 13:11 – BWSEL[2:0]** Bandwidth selection
These bits select the PLL closed loop filter bandwidth, depending on the frequency after the reference divider $F_{PFD}$ as shown in the table below. Selecting the correct filter bandwidth is important to operate the PLL'VCO in its best range.

**Table 17-18.** PLL0 BWSEL Filter

| $F_{PFD}$ | BWSEL[2:0] |
|---|---|
| Reserved | 0b000 |
| 4MHz ≤ $F_{PFD}$ < 10MHz | 0b001 |
| 10MHz ≤ $F_{PFD}$ < 20MHz | 0b010 |
| 20MHz ≤ $F_{PFD}$ < 30MHz | 0b011 |
| 30MHz ≤ $F_{PFD}$ < 60MHz | 0b100 |
| Reserved | 0b101 – 0b111 |


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

**Notes:**

1. $F_{PFD}$ is the frequency of the reference clock divided by the PLL0 reference divider PLLREFDIV.REFDIV. These bits are PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE =1.
2. At elevated temperatures, the effective range of the Bandwidth setting will skew higher. Depending on the input frequency and operating temperature, it may be optimal to change the BWSEL setting to the next higher value.

| Value | Name | Description |
|---|---|---|
| 0x0 | BWSEL0 | TBD |
| 0x1 | BWSEL1 | TBD |
| 0x2 | BWSEL2 | TBD |
| 0x3 | BWSEL3 | TBD |
| 0x4 | BWSEL4 | TBD |
| 0x5 | BWSEL5 | TBD |
| 0x6 | BWSEL6 | TBD |
| 0x7 | BWSEL7 | TBD |

**Bits 10:8 – REFSEL[2:0]** Reference selection

These bits select the PLL0 clock reference, as shown in the table below.

| REFSEL[2:0] | Selectedsource | Description |
|---|---|---|
| 0x0 (2,3) | GCLK | Dedicated GCLK_PLL0_REF clock reference |
| 0x1 | XOSC | XOSC clock reference |
| 0x2 | DFLL48M | DFLL48M clock reference |
| 0x3- 0x7 | n/a | Reserved |

**Notes:**

1. These bits are PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE =1.
2. **IMPORTANT**: If GCLK source is PLL0 then you should not use GCLK_PLL0_REF as input clock source to PLL0. It would create a circular reference, an unstable clock and unexpected behavior.
3. The recommended clock sources for PLL0 are XOSC and the DFLL48M. Using the GLCK as a source is not recommended.

| Value | Name | Description |
|---|---|---|
| 0x0 | GCLK | Dedicated GCLK clock reference |
| 0x1 | XOSC | XOSC clock reference |
| 0x2 | DFLL48M | DFLL48M clock reference |

**Bit 7 – ONDEMAND** On Demand Control

The ONDEMAND operation mode allows the PLL to be enabled or disabled depending on peripheral clock requests.

**Note:** If ONDEMAND is set, the PLL will only be running when requested by a peripheral and enabled (PLLCTRL. ENABLE = 1). If there is no peripheral requesting the PLL's clock source, the PLL will be in a disabled state. If ONDEMAND is disabled, the PLL will always be running when enabled (PLLCTRL.ENABLE = 1). In Standby Sleep mode, the ONDEMAND operation is still active.

This bit is PLLCTRL0.ENABLE protected and cannot be updated if PLLCTRL0.ENABLE = 1.

| Value | Description |
|---|---|
| 0 | The PLL0 is always on. |
| 1 | The PLL0 is running when a peripheral is requesting the PLL to be used as a clock source. The PLL is not running if no peripheral is requesting the PLL clock source. |

**Bit 2 – WRTLOCK** Write Lock

**Note:** Once the WRTLOCK bit is set, it can only be cleared by a reset.


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

| Value | Description |
|---|---|
| 0 | The PLLCTRL, PLLFBDIV, PLLREFDIV and PLLPOSTDIVA/B registers can be modified by a system write. |
| 1 | The PLLCTRL, PLLFBDIV, PLLREFDIV and PLLPOSTDIVA/B registers are write protected, except for bits PLLPOSTDIVA.OUTENn. |

**Bit 1 – ENABLE** PLL Enable

| Value | Description |
|---|---|
| 0 | The PLL0 oscillator is disabled. |
| 1 | The PLL0 oscillator is enabled. |


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

### 17.8.14. PLL Feed-Back Divider

**Name:** PLL0FBDIV
**Offset:** 0x44
**Reset:** 0x00000000
**Property:** RW

**Table 17-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | FBDIV[9:8] | |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

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

**Bits 9:0 – FBDIV[9:0]** PLL Feed-Back Divider Factor
This field determines the ratio of the PLL's VCO output frequency to the PLL Reference input frequency. Writing to the FBDIV bits will cause lock to be lost.
The value of FBDIV, (i.e. PLLFBDIV) must be within the range 21 ≤ FBDIV ≤ 1023.
**Note:** Note: The frequency of the Voltage Controlled Oscillator (VCO) giving the PLL0 oscillation is given by the formula:
FVCO = FCKR * (FBDIV / REFDIV), (i.e., Must be between 800 MHz and 1600 MHz).

$f_{CKR}$ , REFDIV and FBDIV must be selected to satisfy this condition.


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

### 17.8.15. PLL reference divider

**Name:** PLL0REFDIV
**Offset:** 0x48
**Reset:** 0x00000000
**Property:** RW

**Table 17-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | REFDIV[5:0] | | | | | |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:0 – REFDIV[5:0]** PLL reference division factor
This field determines the division factor of the PLL0 input reference frequency. Writing to the REFDIV bits will cause lock to be lost. REFDIV value must be in the range of 1 ≤ REFDIV ≤ 63.
The frequency after the reference divider (FPFD) is given by the formula:
$F_{PFD} = F_{CKR} / REFDIV$ (i.e., $F_{PFD}$ must always be between 4 MHz to 48 MHz.


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

### 17.8.16. PLL output clock divider A

**Name:** PLL0POSTDIVA
**Offset:** 0x4C
**Reset:** 0x20202020
**Property:** RW

**Important:** The PLL0 frequency cannot be changed on the fly while it's the active enabled clock to the system.

**Table 17-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | OUTEN | | POSTDIV[5:0] | | | | | |
| Access | RW | | RW | RW | RW | RW | RW | RW |
| Reset | 0 | | 1 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – OUTEN** PLL output 0 enable

**Bits 5:0 – POSTDIV[5:0]** PLL output 0 clock division factor


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

### 17.8.17. PLL output clock divider B

**Name:** PLL0POSTDIVB
**Offset:** 0x50
**Reset:** 0x00000020
**Property:** RW

**Important:** The PLL0 frequency cannot be changed on the fly while it's the active enabled clock to the system.

**Table 17-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | OUTEN | | POSTDIV[5:0] | | | | | |
| Access | RW | | RW | RW | RW | RW | RW | RW |
| Reset | 0 | | 1 | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – OUTEN** PLL output 4 enable

**Bits 5:0 – POSTDIV[5:0]** PLL output 4 clock division factor


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

### 17.8.18. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x78
**Reset:** 0x00000000
**Property:** R

**Table 17-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | FRACDIV1 | FRACDIV0 | DFLLMUL | DFLLDIFF | DFLLTUNE | DFLLCTRLB | DFLLENABLE | |
| Access | R | R | R | R | R | R | R | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

**Bit 7 – FRACDIV1** FRACDIV1 Synchronization Busy

**Bit 6 – FRACDIV0** FRACDIV0 Synchronization Busy

**Bit 5 – 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.
**Note:** The DFLLMUL synchronization only applies for write operations.

**Bit 4 – DFLLDIFF** DFLLDIFF Synchronization Busy
This bit is cleared when the synchronization of the DFLLDIFF register between the clock domains is complete. This bit is set when the synchronization of the DFLLDIFF register between clock domains is started.
**Note:** The DFLLDIFF synchronization only applies for read operations.

**Bit 3 – DFLLTUNE** DFLLTUNE Synchronization Busy
This bit is cleared when the synchronization of the DFLLTUNE register between the clock domains is complete. This bit is set when the synchronization of the DFLLTUNE register between clock domains is started.


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

**Note:** The DFLLTUNE synchronization applies for read and write operations.

**Bit 2 – DFLLCTRLB** DFLLCTRLB Synchronization Busy
This bit is cleared when the synchronization of the DFLLCTRLB register between the clock domains is complete. This bit is set when the synchronization of the DFLLCTRLB register between clock domains is started.
**Note:** The DFLLCTRLB synchronization only applies for write operations.

**Bit 1 – DFLLENABLE** DFLL48M ENABLE Synchronization Busy
This bit is cleared when the synchronization of the DFLLCTRLA.ENABLE register bit between the clock domains is complete.
**Note:** This bit is set when the synchronization of the DFLLCTRLA.ENABLE register bit between clock domains is started.


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

### 17.8.19. XOSC Calibration Register

**Name:** XOSCCAL
**Offset:** 0x80
**Reset:** 0x00000000
**Property:** RW

**Table 17-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | CAL[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 |
|---|---|---|---|---|---|---|---|---|
| | CAL[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – CAL[15:0]** XOSC Calibration


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

# 18. Generic Clock Controller (GCLK)

## 18.1. Overview

Depending on the application, peripherals may require specific clock frequencies to operate correctly. The Generic Clock controller (GCLK) features 12Generic Clock Generators that can provide a wide range of clock frequencies.

Generators can be set to use different external or internal oscillators as source. The clock of each Generator can be divided according to the user's requirements. The outputs from the Generators are used as sources for the Peripheral Channels, which provide the Generic Clock (GCLK_PERIPH) to the peripheral modules. The number of Peripheral Clocks, GCLK_PERIPH, depends on how many peripherals the device has.
**Note:** Generator 0 is always the direct source of the GCLK_MAIN signal.

## 18.2. Features

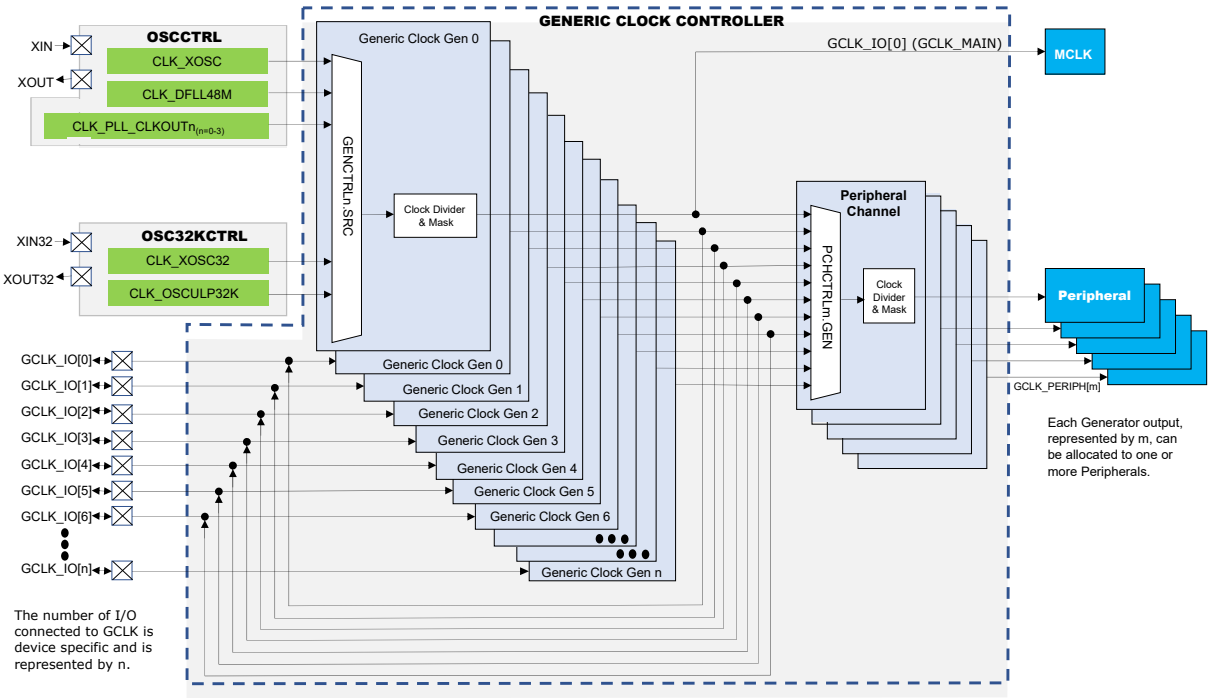
The following are key features of the GCLK module:

- Provides a device-defined, configurable number of Peripheral Channel clocks
- Wide frequency range:
  - Various clock sources
  - Embedded dividers

## 18.3. Block Diagram

The generation of Peripheral Clock signals (GCLK_PERIPH) and the Main Clock (GCLK_MAIN) can be seen in the Device Clocking Diagram below.

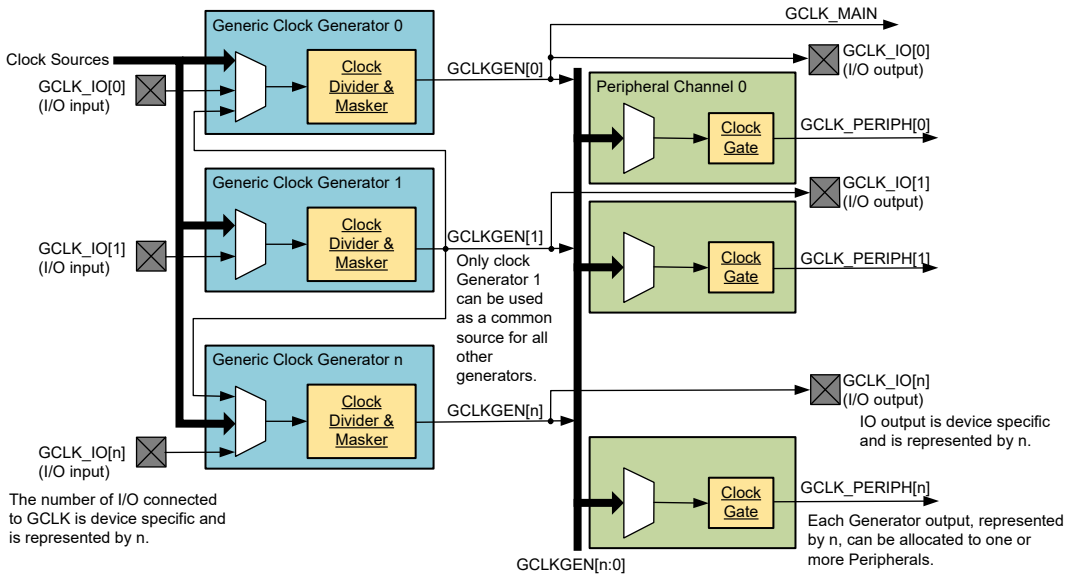
**Figure 18-1.** Device Clocking Diagram




The GCLK block diagram is shown below:


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

**Figure 18-2.** Generic Clock Controller Detailed Block Diagram




## 18.4. Signal Description

**Table 18-1.** GCLK External Signal Descriptions

| Signal Name | Type | Description |
|---|---|---|
| GCLK_IO[n10:0] | Digital I/O | • External Clock source for Generators when input<br>• Generic Clock signal when output |

**Notes:**

1. One signal can be mapped on several pins.
2. The number of I/O connected to GCLK is device specific. The number is 8 for this design.

## 18.5. Functional Description

### 18.5.1. Principle of Operation

The GCLK module is comprised of 12 Generic Clock Generators (Generators) sourcing the 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) to the peripherals.

### 18.5.2. GCLK Basic Operation

#### 18.5.2.1. 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 (GENCTRLn.DIVSEL and GENCTRLn.DIV) by performing a single 32-bit write to the Generator Control register (GENCTRLn).


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

2. The Generic Clock for a peripheral must be configured by writing to the respective Peripheral Channel Control register (PCHCTRLm). 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" (n=0-10), is configured by one dedicated register GENCTRLn.

**Note:** Each Peripheral Channel "m" (m=0-38), is configured by one dedicated register PCHCTRLm.

### 18.5.2.2. 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*.

### 18.5.2.3. Generic Clock Generator

Each Generator (GCLK_GEN) can be set to run from one of 11 different clock sources, GENCTRLn.SRC, except GCLK_GEN[1], which can be set to run from one of 10 sources. GCLK_GEN1 is the only Generator that can be selected as source to others Generators.

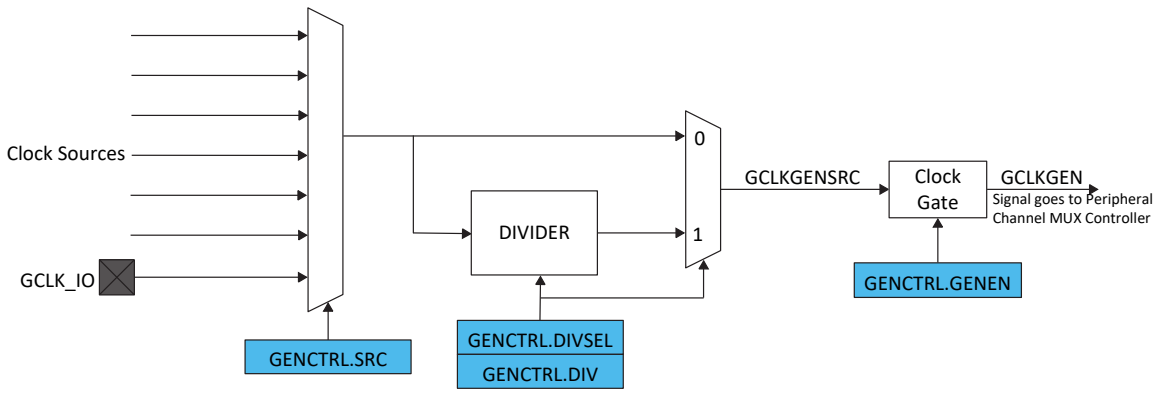
Each Generator GCLK_GEN[x] can be connected to only one specific pin GCLK_IO[x] in a one-to-one GCLK_GENx to GCLK_IOx relationship. A pin GCLK_IO[x] can be set either to act as source to GCLK_GEN[x] or to output the clock signal generated by GCLK_GEN[x].

The selected source can be divided. Each Generator can be enabled or disabled independently.

Each GCLK_GEN clock signal can then be used as the clock source for 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 section for details on synchronous clock generation.

**Figure 18-3.** Generic Clock Generator




### 18.5.2.4. Enabling a Generator

Users can enable the generator by writing a '1' to the Generator Enable bit in the Generator Control register (GENCTRLn.GENEN = 1).

### 18.5.2.5. Disabling a Generator

Users can disable the generator by writing a '0' to GENCTRLn.GENEN. When GENCTRLn.GENEN = 0, the GCLK_GEN[n] clock is disabled and gated off.


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

#### 18.5.2.6. 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, from clock source 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 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. Accordingly the SYNCBUSY.GENCTRLn bit in the SYNCBUSY register will remain '1' until the switch operation is completed.

The available clock sources are device dependent (usually the oscillators, RC oscillators, PLL). Only clock Generator 1 can be used as a common source for all other generators.

#### 18.5.2.7. 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 dependent 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.

#### 18.5.2.8. 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.

#### 18.5.2.9. 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/IOn) must first be configured as output by writing the corresponding PORT registers.

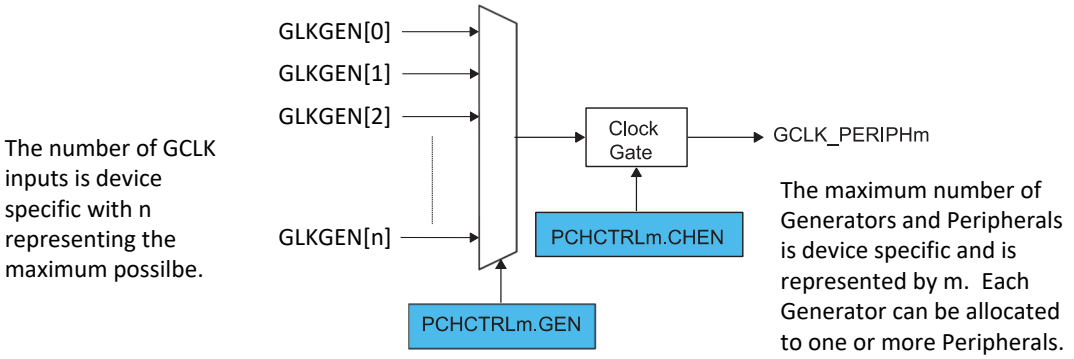
If GENCTRLn.OE is ‘0’, the according I/O pin is set to an Output Off Value, which is selected by GENCTRLn.OOV. If GENCTRLn.OOV is '0', the output clock will be low. If this bit is '1', the output clock will be high.

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 ‘0’. If GENCTRLn.RUNSTDBY is '1', the GCLKGEN clock is kept running and output to the I/O pin.


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

### 18.5.3. Peripheral Clock

**Figure 18-4.** Peripheral Clock




#### 18.5.3.1. 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.

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. The PCHCTRLm.CHEN bit will continue to read as its previous state until the synchronization is complete.

#### 18.5.3.2. 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.

#### 18.5.3.3. Selecting the Clock Source for a Peripheral

When changing a peripheral clock source by writing to PCHCTRLm.GEN, 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. Assert 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 as '1'.

#### 18.5.3.4. Configuration Lock

The peripheral clock configuration can be locked from 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.

The generator 0 is one exception concerning the configuration lock. 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.


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

In case of an external Reset, 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.

## 18.5.4. GCLK Additional Features

### 18.5.4.1. 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.

## 18.5.5. Sleep Mode Operation

### 18.5.5.1. 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 that pin. When set, the GCLK can continuously output the generator output to GCLK_IO. Refer to *External Clock* for additional information.

### 18.5.5.2. Minimize Power Consumption in Standby

The following table identifies when a Clock Generator is off in Standby mode, minimizing the power consumption:

**Table 18-2.** 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 |

### 18.5.5.3. Entering Standby Mode

There may occur a delay when the device is put into Standby mode, until all clocks are gated 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 off properly. The duration of this verification is frequency-dependent.

For further information, refer to the *Power Manager* chapter.

## 18.5.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some registers must 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 (polling method) and to assert glitch free internal operation. Changing the bit value under ongoing synchronization will *not* generate an error.

The following registers are synchronized when written:

- The Generic Clock Generator Control register (GENCTRLn)
- The Control A register (CTRLA)

Required write synchronization is denoted by the “Write Synchronized” property in the register description.

## 18.6. Peripheral Dependencies

**Table 18-3.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| GCLK | 0x44008000 | | **APB:**<br>CLKMSK[0]<br>MASK8 | --- | 8<br>STATUS0<br>PERID8 | | | PD_CORE_SW |

## 18.7. GCLK Peripheral Channel Control Mapping

**Table 18-4.** GCLK Peripheral Channel Control Mapping

| Peripheral Channel Control Register | Peripheral:Clock |
|---|---|
| PCHCTRL[0] | OSCCTRL:DFLL48M |
| PCHCTRL[1] | OSCCTRL:PLL |
| PCHCTRL[2] | FREQM:MSR0 |
| PCHCTRL[3] | FREQM:MSR1 |
| PCHCTRL[4] | FREQM:REF |
| PCHCTRL[5] | EIC |
| PCHCTRL[6] | CAN0 |
| PCHCTRL[7] | CAN1 |
| PCHCTRL[8] | EVSYS:CH0 |
| PCHCTRL[9] | EVSYS:CH1 |
| PCHCTRL[10] | EVSYS:CH2 |
| PCHCTRL[11] | EVSYS:CH3 |
| PCHCTRL[12] | EVSYS:CH4 |
| PCHCTRL[13] | EVSYS:CH5 |
| PCHCTRL[14] | EVSYS:CH6 |
| PCHCTRL[15] | EVSYS:CH7 |
| PCHCTRL[16] | EVSYS:CH8 |
| PCHCTRL[17] | EVSYS:CH9 |
| PCHCTRL[18] | EVSYS:CH10 |
| PCHCTRL[19] | EVSYS:CH11 |


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

**Table 18-4.** GCLK Peripheral Channel Control Mapping (continued)

| Peripheral Channel Control Register | Peripheral:Clock |
|---|---|
| PCHCTRL[20] | SERCOM0:SLOW<br>SERCOM1:SLOW<br>SERCOM2:SLOW<br>SERCOM3:SLOW |
| PCHCTRL[21] | SERCOM0:CORE |
| PCHCTRL[22] | SERCOM1:CORE |
| PCHCTRL[23] | SERCOM2:CORE |
| PCHCTRL[24] | SERCOM3:CORE |
| PCHCTRL[25] | TCC0:TCC01<br>TCC1:TCC01 |
| PCHCTRL[26] | TCC2 |
| PCHCTRL[27] | TCC3 |
| PCHCTRL[28] | SERCOM4:CORE |
| PCHCTRL[29] | SERCOM5:CORE |
| PCHCTRL[30] | TCC4 |
| PCHCTRL[31] | TCC5 |
| PCHCTRL[32] | TCC6 |
| PCHCTRL[33] | ADC |
| PCHCTRL[34] | AC |
| PCHCTRL[35] | PTC |
| PCHCTRL[36] | CCL0 |
| PCHCTRL[37] | CCL1 |
| PCHCTRL[38] | USB |


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

## 18.8. 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 | GENCTRL[5:0] | | | | | | | SWRST |
| | | 15:8 | | | GENCTRL[11:6] | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08<br>...<br>0x1F | Reserved | | | | | | | | | |
| 0x20 | GENCTRL0 | 7:0 | | | | SRC[4:0] | | | | |
| | | 15:8 | | | RUNSTDBY | DIVSEL | OE | OOV | IDC | GENEN |
| | | 23:16 | DIV[7:0] | | | | | | | |
| | | 31:24 | DIV[15:8] | | | | | | | |
| ... | | | | | | | | | | |
| 0x5C | GENCTRL15 | 7:0 | | | | SRC[4:0] | | | | |
| | | 15:8 | | | RUNSTDBY | DIVSEL | OE | OOV | IDC | GENEN |
| | | 23:16 | DIV[7:0] | | | | | | | |
| | | 31:24 | DIV[15:8] | | | | | | | |
| 0x60<br>...<br>0x7F | Reserved | | | | | | | | | |
| 0x80 | PCHCTRL0 | 7:0 | WRTLOCK | CHEN | | | GEN[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| ... | | | | | | | | | | |
| 0x0118 | PCHCTRL38 | 7:0 | WRTLOCK | CHEN | | | GEN[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 18.8.1. Control

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** RW

**Table 18-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SWRST |
| Access | | | | | | | | RW |
| 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 any 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.

### 18.8.2. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** R

**Table 18-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | GENCTRL[11:6] | | | | | |
| Access | | | R | R | R | R | R | R |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

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

**Bits 13:2 – GENCTRL[11:0]** Generic Clock Generator Control x Synchronization Busy bit

**Bit 0 – SWRST** Software Reset Synchronization Busy bit
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.

### 18.8.3. Generic Clock Generator Control

**Name:** GENCTRL
**Offset:** 0x20 + n*0x04 [n=0..15]
**Reset:** 0x00000000
**Property:** RW

**Table 18-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIV[15:8] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | DIV[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 |
|---|---|---|---|---|---|---|---|---|
| | | | RUNSTDBY | DIVSEL | OE | OOV | IDC | GENEN |
| Access | | | RW | RW | RW | RW | RW | RW |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

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

**Bits 31:16 – DIV[15:0]** Division Factor
These bits represent a division value for the corresponding GCLK Generator input source clock defined by GENCTRLn.SRC. The actual division factor is dependent on the state of DIVSEL.

**Table 18-11.** Division Factor Bits

| Generic Clock Generator | Division Factor Bits |
|---|---|
| Clock Generator [15:0] | 16 division factor bits - DIV[16:0] |

**Notes:**
If GENCTRLn.DIVSEL = 0 then FGCLK = GENCTRLn.SRC/DIV

- If GENCTRLn.DIV is an odd number, then GENCTRLn.IDC should be set to IDC = 1
- If GENCTRLn.DIV is an even number, then GENCTRLn.IDC should be set to IDC = 0

If GENCTRLn.DIVSEL = 1 then FGCLK = GENCTRLn.SRC/2^(DIV+1)

- GENCTRLn.IDC should ALWAYS be set to IDC = 0

**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_IO pin. If GENCTRLn.OE is zero, this bit has no effect and the generator will only be running if a peripheral requires the clock.


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

| Value | Description |
|---|---|
| 0 | The Generator is stopped in Standby and the GCLK_IO 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_IO 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.

| Value | Name | Description |
|---|---|---|
| 0 | DIV1 | Divide input directly by divider factor |
| 1 | DIV2 | Divide input by 2^(divider factor+ 1) |

**Bit 11 – OE** Output Enable

This bit is used to output the Generator clock output to the corresponding pin (GCLK_IO), as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

| Value | Description |
|---|---|
| 0 | No Generator clock signal on pin GCLK_IO. |
| 1 | The Generator clock signal is output on the corresponding GCLK_IO, unless GCLK_IO 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) when the Generator is turned off or the OE bit is zero, as long as GCLK_IO is not defined as the Generator source in the GENCTRLn.SRC bit field.

| Value | Description |
|---|---|
| 0 | The GCLK_IO will be LOW when generator is turned off or when the OE bit is zero. |
| 1 | The GCLK_IO 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.
**Note:** If DIVSEL = 1 this bit should always be set to IDC = 0.
If DIVSEL = 0 and DIV = odd number then IDC=1, else if DIV = even number IDC = 0.

| 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** Generic Clock Generator Enable

This bit is used to enable and disable the Generator.

| Value | Description |
|---|---|
| 0 | Generator is disabled. |
| 1 | Generator is enabled. |

**Bits 4:0 – SRC[4:0]** Source Select

These bits select the Generator clock source, as shown in this table.

**Table 18-8.** Generator Clock Source Selection

| Value (GENCTRLn.SRC) | Name | Description |
|---|---|---|
| 0x00 | XOSC | XOSC Crystal/Clock Oscillator |
| 0x01 | GCLK_IOn | Generator I/O input pin (GCLK_IOn) where n corresponds to the Clock Generator number. |


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

**Table 18-8.** Generator Clock Source Selection (continued)

| Value (GENCTRLn.SRC) | Name | Description |
|---|---|---|
| 0x02 | GCLK_GEN1 | Generic clock generator 1 (GCLK1) |
| 0x03 | OSCULP32K (32.768 KHz) | Internal Ultra-Low Power 32K RC Oscillator |
| 0x04 | XOSC32K (32.768 KHz) | 32 KHz Crystal Oscillator |
| 0x05 | DFLL48M (RC48M) | Internal 48 MHz RC Oscillator |
| 0x06 | PLL0_CLKOUT1 | Digital Phase Lock Loop, PLL0 Output 1 |
| 0x07 | PLL0_CLKOUT2 | Digital Phase Lock Loop, PLL0 Output 2 |
| 0x08 | PLL0_CLKOUT3 | Digital Phase Lock Loop, PLL0 Output 3 |
| 0x09 | PLL0_CLKOUT4 | Digital Phase Lock Loop, PLL0 Output 4 |
| 0x0A | PLL0_CLKOUT5 | Digital Phase Lock Loop, PLL0 Output 5 |
| 0x0B-0x1F | Reserved | Reserved |

**Note:** GENCTRL1.SRC = 0x2 is invalid for GCLK_GEN1 only.

Any reset will reset all GENCTRLn registers. The Reset values of the GENCTRLn registers are shown in table below.

**Table 18-9.** GENCTRLn Reset Value after a Power Reset

| GCLK Generator | Reset Value after a Power Reset |
|---|---|
| GCLK0 (GENCTRL0) | GCLK.GENCTRL0 = 0x00000105 (DFLL48M, Internal 48 MHz RC Oscillator, GCLK0 Enabled) |
| GCLK1 (GENCTRL1) - GCLK11 (GENCTRL11) | GCLK.GENCTRL1- GCLK.GENCTRL11 = 0x0000000 (XOSC, GCLK1-GCLK11 disabled) |

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 18-10.** GENCTRLn Reset Value after a User Reset

| GCLK Generator | Reset Value after a User Reset |
|---|---|
| GCLK0 (GENCTRL0) | GCLK.GENCTRL0 = 0x00000105 (DFLL48M, Internal 48MHz RC Oscillator, GCLK0 Enabled) |
| GCLK1 (GENCTRL1) - GCLK1 (GENCTRL11) | GCLK.GENCTRL1- GCLK.GENCTRL11 = No change if the generator is used by a Peripheral Channel m with PCHCTRLm.WRTLOCK = 1<br>else 0x00000000 |


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

### 18.8.4. Peripheral Clock Control

**Name:** PCHCTRL
**Offset:** 0x80 + n*0x04 [n=0..38]
**Reset:** 0x00000000
**Property:** RW

PCHCTRLm controls the settings of Peripheral Channel number m (m=0..38).

**Table 18-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | RW | RW | | | RW | RW | RW | RW |
| 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]** Generic Clock Generator
This bit field selects the Generator to be used as the source of a peripheral clock, as shown in the table below:


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

**Table 18-13.** Generator Selection

| Value | Description |
|---|---|
| 0x0 | Generic Clock Generator 0 |
| 0x1 | Generic Clock Generator 1 |
| 0x2 | Generic Clock Generator 2 |
| 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 |
| 0xC - 0xF | Reserved |

**Table 18-14.** Reset Value after a User Reset or a Power Reset

| Reset | PCHCTRLm.GEN | PCHCTRLm.CHEN | PCHCTRLm.WRTLOCK |
|---|---|---|---|
| Power Reset | 0x0 | 0x0 | 0x0 |
| User Reset | 0x0 | 0x0 | 0x0 |

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.

| Value | Name | Description |
|---|---|---|
| 0x0 | GCLK0 | Generic clock generator 0 |
| 0x1 | GCLK1 | Generic clock generator 1 |
| 0x2 | GCLK2 | Generic clock generator 2 |
| 0x3 | GCLK3 | Generic clock generator 3 |
| 0x4 | GCLK4 | Generic clock generator 4 |
| 0x5 | GCLK5 | Generic clock generator 5 |
| 0x6 | GCLK6 | Generic clock generator 6 |
| 0x7 | GCLK7 | Generic clock generator 7 |
| 0x8 | GCLK8 | Generic clock generator 8 |
| 0x9 | GCLK9 | Generic clock generator 9 |
| 0xA | GCLK10 | Generic clock generator 10 |
| 0xB | GCLK11 | Generic clock generator 11 |
| 0xC | GCLK12 | Generic clock generator 12 |
| 0xD | GCLK13 | Generic clock generator 13 |
| 0xE | GCLK14 | Generic clock generator 14 |
| 0xF | GCLK15 | Generic clock generator 15 |


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

# 19. Main Clock (MCLK)

## 19.1. Overview

The Main Clock (MCLK) controls the generation of the device synchronous clocks.

Using a clock provided by the Generic Clock Generator 0 Module (GCLK_MAIN), the MCLK provides synchronous clocks to the CPU, the three Advanced Peripheral Buses (APB), the Advanced High-performance Buses (AHB), and the peripherals connected to these buses.

The synchronous system clocks are divided into 3 clock domains, each can run at different frequencies. The generated clocks can be masked for individual modules, enabling the user to minimize power consumption.

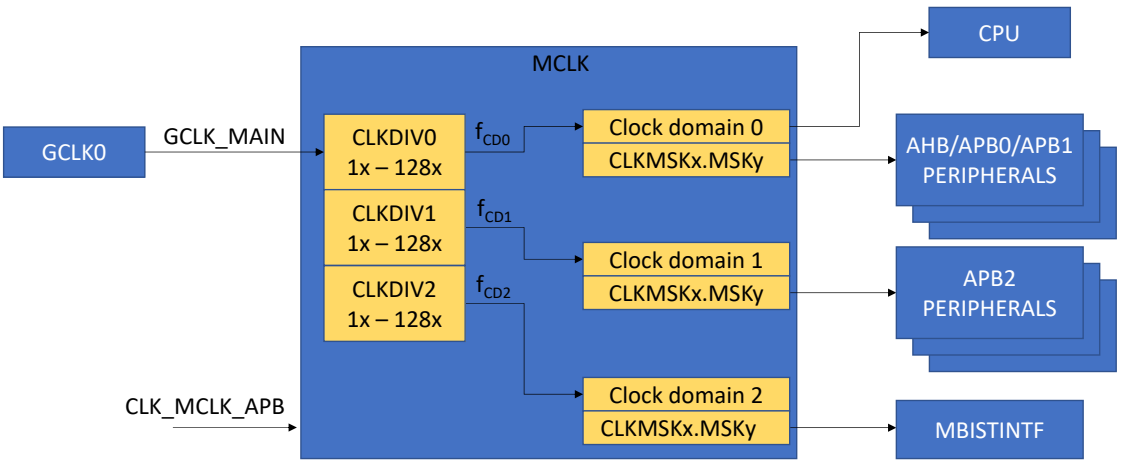
## 19.2. Features

- **Clock source:** GCLK_MAIN derived from GCLK0
- Three clock domains:
  - 1, 2, 4, 8, 16, 32, 64, 128 independent division factors from GCLK_MAIN
  - **Clock domain 0:** Generates individual clocks running at fCD0 frequency for CPU, AHB, APB0/1, and all peripherals connected to them(1)
  - **Clock domain 1:** Generates individual clocks running at fCD1 frequency for APB2 and all peripherals connected to it (1)
  - **Clock domain 2:** Generates a clock running at fCD2 dedicated for the MBISTINTF module (1)
- Clock domain divider update on the fly
- Peripheral-level clock gating through individual mask bits

**Note:**

1. Refer to the Peripherals Configuration Summary for the list of peripherals and the associated clock domain, and clock mask register and index.

## 19.3. Block Diagram




**Note:** Specific CLKMSKx and MSKy indices can be found in *MCLK AHB/APB Clock Enable Peripheral Mapping*.


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

## 19.4. Functional Description

### 19.4.1. Principle of Operation

The GCLK_MAIN clock signal from the GCLK0 module is the source for the MCLK, which in turn is the common root for the synchronous clocks for the CPU, APB0/1/2 (Advanced Peripheral Bus), and the AHB (Advanced High-performance Bus).

For each clock domain, the CLKDIVx.DIV bitfield allows selecting a power of 2 divided clock (from 1 to 128) from the GCLK_MAIN clock, ensuring synchronous clock sources for each clock domain. The default configuration after reset and ROM code execution depends on the user configuration of BOOTCFG1/1A.MCLKDIVx registers.

### 19.4.2. Basic Operation

#### 19.4.2.1. Enabling, Disabling, and Resetting

The MCLK module is always enabled and cannot be reset.

#### 19.4.2.2. Selecting the Synchronous Clock Division Ratio

Each CLKDIVx.DIV divider can be changed on the fly without halting or disabling the peripheral modules connected to it. This change allows responding to variable load in the application and is applied to all synchronous clocks belonging to the corresponding clock domain at the same time.

To ensure correct operation:

- Frequencies must be selected so that fCD0 ≥ fCD1 ≥ fCD2. Values that violates this relationship will be discarded and will generate a bus error, logged in the PAC module.
- Frequencies must never exceed the specified maximum frequency for each clock domain given in the electrical characteristics specifications.
- Wait for INTFLAG.CLKRDY = 1, change CLKDIVx.DIV, and wait for CLKRDY to come back to 1.

#### 19.4.2.3. Clock Ready Flag

There is a slight delay between writing to CLKDIVx.DIV 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.

#### 19.4.2.4. Peripheral Clock Masking

All synchronous clocks can then be masked/unmasked individually by writing the corresponding bit in the Clock Mask registers (CLKMSKx.MASKy), where the x and y indexes are defined in the *Peripherals Configuration Summary*. To find the default state of each peripheral clock, consult the associated Clock Mask register reset value.

All synchronous clocks are unmasked by default after reset.

When a peripheral clock is unmasked and the peripheral is:

- Not enabled (CTRLA.ENABLE = 0), then the clock is automatically gated at the MCLK output to save power consumption, and automatically ungated when read/write register accesses are made in this peripheral
- Enabled (CTRLA.ENABLE = 1), the clock is permanently requested and delivered

When a peripheral clock is masked, its synchronous clock is gated at the MCLK level, saving power consumption in the clock tree, but making the peripheral registers no more accessible. Read/write accesses to the peripheral registers will fail and return an error.

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


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

the APB0 bridge, will make it impossible to write the mask registers again. In this case, they can only be re-enabled by a system reset.

A module may be connected to several clock domains (for instance, AHB and APB), in which case it will have several mask bits.

The CPU clock to the Arm Cortex processor is derived from the clock domains 0 and runs at fCD0. It is controlled automatically and does not depend on a CLKMSKx.MASK bit.

### 19.4.3. Interrupts

The peripheral has the following interrupt sources:

- **Clock Ready (CKRDY):** indicates that the programmed divider setting in any CLKDIVx.DIV has taken effect. This interrupt is a synchronous wake-up source.

### 19.4.4. Sleep Mode Operation

In Active, Idle, and Sleepwalking modes, the MCLK will automatically gate any clock domain where all peripherals are not requesting their clocks.

In Standby Sleep mode, the MCLK is frozen if no synchronous clock is required.

### 19.4.5. 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.

## 19.5. Peripheral Dependencies

**Table 19-1.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| MCLK | 0x4400A000 | 9 : CKRDY | **APB:** CLKMSK[0] MASK9 | --- | 9 STATUS0 PERID9 | | | PD_CORE_SW |


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

## 19.6. MCLK AHB/APB Clock Enable Peripheral Mapping

**Table 19-2.** MCLK AHB/APB Clock Enable Peripheral Mapping

| Peripheral BUS Clock Enable Mask Register | Clock Enable Mask Bits | Peripheral | Bus |
|---|---|---|---|
| CLKMSK[0] | MASK0 | DSU | AHB |
| | MASK1 | FCR | AHB |
| | MASK2 | FCW | AHB |
| | MASK3 | PM | APB |
| | MASK4 | SUPC | APB |
| | MASK5 | RSTC | APB |
| | MASK6 | OSCCTRL | APB |
| | MASK7 | OSC32KCTRL | APB |
| | MASK8 | GCLK | APB |
| | MASK9 | MCLK | APB |
| | MASK10 | FREQM | APB |
| | MASK11 | WDT | APB |
| | MASK12 | RTC | APB |
| | MASK13 | EIC | APB |
| | MASK14 | PAC | APB |
| | MASK15 | TRAM | APB |
| | MASK21 | MCRAMC | APB |
| | MASK22 | MCRAMC | AHB |
| | MASK23 | CAN0 | AHB |
| | MASK24 | CAN1 | AHB |
| | MASK25 | H2PB0 | AHB |
| | MASK26 | H2PB0 | APB |
| | MASK27 | PORT | APB |
| | MASK28 | DMAC | AHB |
| | MASK29 | DMAC | APB |
| | MASK30 | HMATRIX2 | AHB |
| | MASK31 | HMATRIX2 | APB |


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

**Table 19-2.** MCLK AHB/APB Clock Enable Peripheral Mapping (continued)

| Peripheral BUS Clock Enable Mask Register | Clock Enable Mask Bits | Peripheral | Bus |
|---|---|---|---|
| CLKMSK[1] | MASK0 | BROMC | AHB |
| | MASK1 | BROMC | APB |
| | MASK2 | IDAU | APB |
| | MASK3 | EVSYS | APB |
| | MASK4 | SERCOM0 | APB |
| | MASK5 | SERCOM1 | APB |
| | MASK6 | SERCOM2 | APB |
| | MASK7 | SERCOM3 | APB |
| | MASK8 | TCC0 | APB |
| | MASK9 | TCC1 | APB |
| | MASK10 | TCC2 | APB |
| | MASK11 | TCC3 | APB |
| | MASK12 | SERCOM4 | APB |
| | MASK13 | SERCOM5 | APB |
| | MASK14 | TCC4 | APB |
| | MASK15 | TCC5 | APB |
| | MASK16 | TCC6 | APB |
| | MASK17 | ADC | APB |
| | MASK18 | AC | APB |
| | MASK19 | PTC | APB |
| | MASK20 | CCL0 | APB |
| | MASK21 | CCL1 | APB |
| | MASK22 | USB | AHB |
| | MASK23 | USB | APB |
| | MASK25 | H2PB1 | AHB |
| | MASK26 | H2PB1 | APB |
| | MASK27 | H2PB2 | AHB |
| | MASK28 | H2PB2 | APB |


© 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 | | | | | | | | CKRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | INTENSET | 7:0 | | | | | | | | CKRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | INTFLAG | 7:0 | | | | | | | | CKRDY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | CLKDIV0 | 7:0 | DIV[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | CLKDIV1 | 7:0 | DIV[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | CLKDIV2 | 7:0 | DIV[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 ... 0x3B | Reserved | | | | | | | | | |
| 0x3C | CLKMSK0 | 7:0 | MASK7 | MASK6 | MASK5 | MASK4 | MASK3 | MASK2 | MASK1 | MASK0 |
| | | 15:8 | MASK15 | MASK14 | MASK13 | MASK12 | MASK11 | MASK10 | MASK9 | MASK8 |
| | | 23:16 | MASK23 | MASK22 | MASK21 | MASK20 | MASK19 | MASK18 | MASK17 | MASK16 |
| | | 31:24 | MASK31 | MASK30 | MASK29 | MASK28 | MASK27 | MASK26 | MASK25 | MASK24 |
| 0x40 | CLKMSK1 | 7:0 | MASK7 | MASK6 | MASK5 | MASK4 | MASK3 | MASK2 | MASK1 | MASK0 |
| | | 15:8 | MASK15 | MASK14 | MASK13 | MASK12 | MASK11 | MASK10 | MASK9 | MASK8 |
| | | 23:16 | MASK23 | MASK22 | MASK21 | MASK20 | MASK19 | MASK18 | MASK17 | MASK16 |
| | | 31:24 | MASK31 | MASK30 | MASK29 | MASK28 | MASK27 | MASK26 | MASK25 | MASK24 |
| 0x44 | CLKMSK2 | 7:0 | MASK7 | MASK6 | MASK5 | MASK4 | MASK3 | MASK2 | MASK1 | MASK0 |
| | | 15:8 | MASK15 | MASK14 | MASK13 | MASK12 | MASK11 | MASK10 | MASK9 | MASK8 |
| | | 23:16 | MASK23 | MASK22 | MASK21 | MASK20 | MASK19 | MASK18 | MASK17 | MASK16 |
| | | 31:24 | MASK31 | MASK30 | MASK29 | MASK28 | MASK27 | MASK26 | MASK25 | MASK24 |


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

### 19.7.1. Interrupt Enable Clear Register

**Name:** INTENCLR
**Offset:** 0x0
**Reset:** 0x00000000
**Property:** R/K

**Table 19-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CKRDY |
| Access | | | | | | | | R/K |
| Reset | | | | | | | | 0 |

**Bit 0 – CKRDY** Clock Ready

**Note:** Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.

| Value | Description |
|---|---|
| 1 | The Clock Ready interrupt is enabled. |
| 0 | The Clock Ready interrupt is disabled |


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

### 19.7.2. Interrupt Enable Set Register

**Name:** INTENSET
**Offset:** 0x4
**Reset:** 0x00000000
**Property:** R/S

**Table 19-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CKRDY |
| Access | | | | | | | | R/S |
| Reset | | | | | | | | 0 |

**Bit 0 – CKRDY**

Writing a '1' to this bit will set the Clock Ready Interrupt Enable bit and enable the Clock Ready interrupt.

| Value | Description |
|---|---|
| 1 | The Clock Ready interrupt is enabled. |
| 0 | The Clock Ready interrupt is disabled. |


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

### 19.7.3. Interrupt Flag Status Register

**Name:** INTFLAG
**Offset:** 0x8
**Reset:** 0x00000001
**Property:** K

**Note:** Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

**Table 19-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CKRDY |
| Access | | | | | | | | HS/K |
| Reset | | | | | | | | 1 |

**Bit 0 – CKRDY** Clock Ready
This flag is set when the system clocks have frequencies as indicated in the CLKDIVx registers and will generate an interrupt if CKRDY interrupt enable is set to '1'.
**Notes:**

1. Writing a '1' to this bit will clear the Clock Ready Interrupt Enable bit and the corresponding interrupt request.
2. This flag is **set by hardware** when the system clocks have frequencies as indicated in the CLKDIVx registers and will generate an interrupt if CKRDY interrupt enable is set to '1'.

| 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.

### 19.7.4. Clock Divider Control n Register

**Name:** CLKDIV
**Offset:** 0x0C + n*0x04 [n=0..2]
**Reset:** 0x00000001
**Property:** R/W

**Table 19-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | 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]** Division Ratio
These bits define the division ratio of the main clock prescaler related to the Clock Domain controlled by the CLKDIVn register. To ensure correct operation, frequencies must be selected so that FCLKDIV[n]≥ FCLKDIV[n+1] (i.e. CLKDIVn.DIV ≤ CLKDIV(n+1).DIV).
Frequencies must never exceed the specified maximum frequency for each clock domain. An update to this register may not take affect immediately. The MCLK peripheral will wait for the falling edge of the previous clock and the new clock to coincide before switching. The INTFLAG.CKRDY can be used to determine when MCLK has made the switch.
**Note:** All other values are reserved or invalid.

| Value | Name | Description |
|---|---|---|
| 1 | DIV1 | Divide by 1 |
| 2 | DIV2 | Divide by 2 |
| 4 | DIV4 | Divide by 4 |
| 8 | DIV8 | Divide by 8 |
| 16 | DIV16 | Divide by 16 |
| 32 | DIV32 | Divide by 32 |
| 64 | DIV64 | Divide by 64 |
| 128 | DIV128 | Divide by 128 |


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

### 19.7.5. Peripheral Clock Enable Mask n Register

**Name:** CLKMSK
**Offset:** 0x3C + n*0x04 [n=0..2]
**Reset:** 0x00000000
**Property:** R/W

**Table 19-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | MASK31 | MASK30 | MASK29 | MASK28 | MASK27 | MASK26 | MASK25 | MASK24 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | MASK23 | MASK22 | MASK21 | MASK20 | MASK19 | MASK18 | MASK17 | MASK16 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | MASK15 | MASK14 | MASK13 | MASK12 | MASK11 | MASK10 | MASK9 | MASK8 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | MASK7 | MASK6 | MASK5 | MASK4 | MASK3 | MASK2 | MASK1 | MASK0 |
| Access | R/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 – MASK** Peripheral Clock Mask bit

| Value | Description |
|---|---|
| 0 | Peripheral clock stopped |
| 1 | Peripheral clock enabled |


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

# 20. 32 KHz Oscillators Controller (OSC32KCTRL)

## 20.1. Overview

The 32 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)

The OSC32KCTRL sub-peripherals can be enabled, disabled, and monitored through interface registers. All sub-peripheral statuses are collected in the Status register (STATUS). They can additionally trigger interrupts upon status changes through the INTENSET, INTENCLR, and INTFLAG registers. The OSC32KCTRL provides clock sources to the Generic Clock Controller (GCLK), Real-Time Counter (RTC), Watchdog Timer (WDT), and Reset Power Management Unit (RPMU).

## 20.2. Features

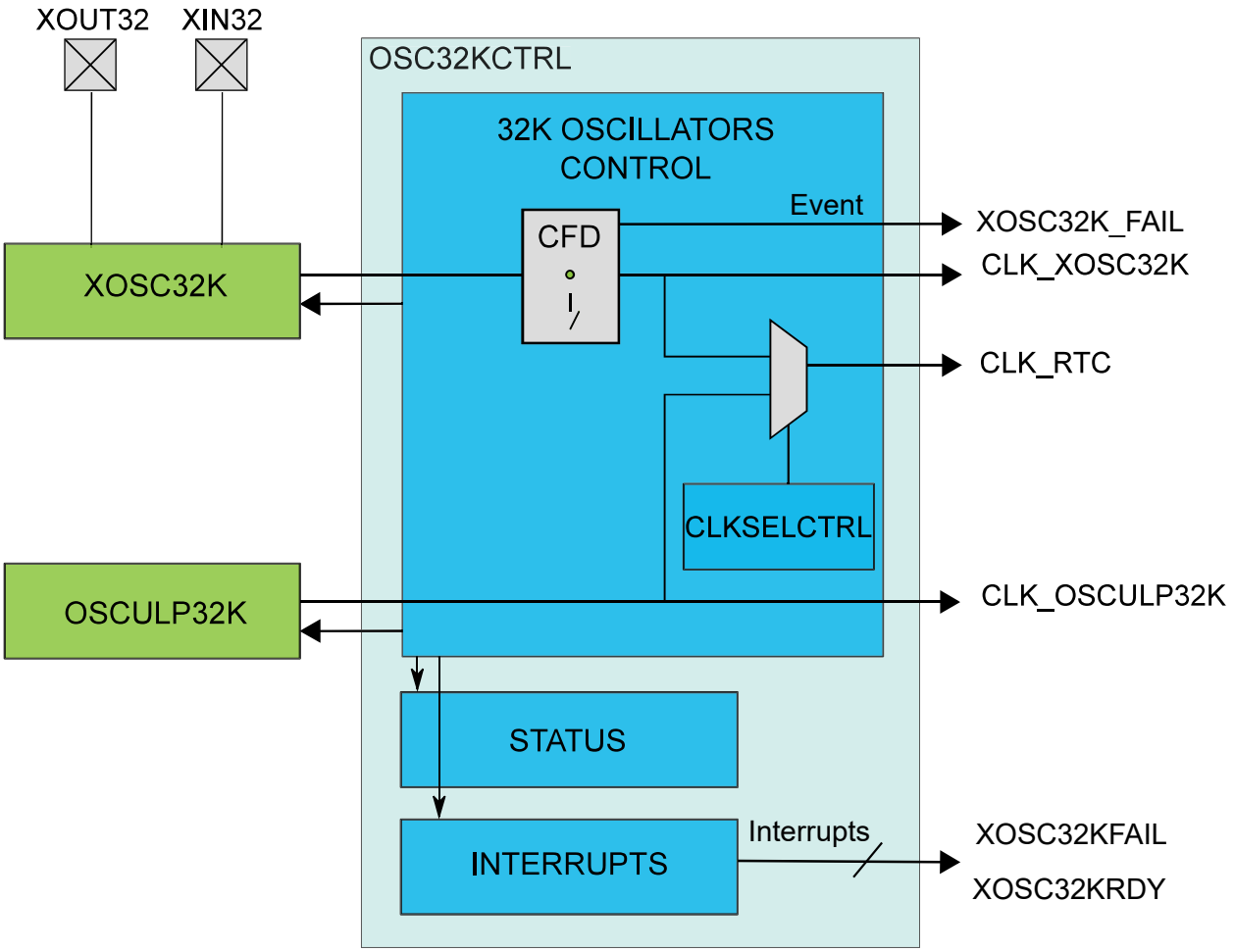
The following are key features of the OSC32KCTRL module:

- 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
- 1.024 kHz clock outputs available


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

## 20.3. Block Diagram

**Figure 20-1.** OSC32KCTRL Block Diagram




## 20.4. 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 and do not need to be configured by the user. The external 32.768 kHz crystal must be connected between the XIN32 and XOUT32 pins, along with any required load capacitors.

## 20.5. Functional Description

### 20.5.1. Principle of Operation

The XOSC32K and OSCULP32K are configured through the OSC32KCTRL Control registers. Through this interface, the sub-peripherals are enabled, disabled, or have their calibration values updated.

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.


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

### 20.5.2. 32 kHz External Crystal Oscillator (XOSC32K) Operation

The XOSC32K can operate in the following two 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 (XOSC32K.XTALEN = 1), the XIN32 and XOUT32 pins are controlled by XOSC32K.ENABLE, and the GPIO functions are overridden on both pins. When in external clock mode (XOSC32K.XTALEN = 0), only the XIN32 pin will be overridden and controlled by XOSC32K.ENABLE, 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). It is necessary to wait for STATUS.XOSC32KRDY bit to set, before disabling the XOSC32K. The XOSC32K is disabled by writing a '0' to the Enable bit in the 32 kHz External Crystal Oscillator Control register (XOSC32K.ENABLE = 0). It is necessary to wait for STATUS.XOSC32KRDY to clear before enabling the XOSC32K.

**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. This bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored.

**Gain Selection:** When a crystal oscillator is selected, a user controllable gain is provided through the CGM control bits. The user must review the electrical specification. This bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored.

**Startup:** XOSC32K.STARTUP[3:0] selects the startup time. It is configurable and applicable for crystal mode or external clock mode. These bits must be written when XOSC32K.ENABLE = 0. Otherwise, the write on these bits is ignored.

The XOSC32K will behave differently in different sleep modes based on the settings of XOSC32K.ONDEMAND and XOSC32K.ENABLE bits. The XOSC32K.ONDEMAND bit must be written when XOSC32K.ENABLE = 0. Otherwise, the write on this bit is ignored. If XOSC32K.ENABLE = 0, the XOSC32K will be always stopped. For XOS32K.ENABLE = 1, following table is valid:

| XOSC32K.ONDEMAND | Sleep Behavior |
|---|---|
| 0 | Always run |
| 1 | Run if requested by a peripheral |

As a crystal oscillator usually requires a long start-up time, the 32 kHz External Crystal Oscillator will keep running across resets, 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 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 (XOSC32K.STARTUP[3:0]) 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).

### 20.5.3. 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


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

XOSC32K clock with reduced latency and allows to switch 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 pre-scaler. This allows to configure the safe clock to fulfill the operative conditions of the microcontroller.

In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral.

The Oscillator's registers allow to enable, disable, and configure the CFD. The Status register provides status flags on failure and clock switch conditions. The CFD can be configured to trigger an interrupt or an event when a failure is detected.

**Clock Failure Detection**

The CFD is reset only at Power-on Reset (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). The 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 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. If the Event Output enable bit in the Event Control register (EVCTRL.CFDEO) is set, an output event is generated.

After a clock failure is issued, the CFD continuously monitors of the XOSC32K clock 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 to maintain an active clock during the XOSC32K clock failure. The safe clock source is the OSCULP32K oscillator clock. Both 32 kHz and 1 kHz outputs of the XOSC32K are replaced by the OSCULP32K, 32 kHz and 1 kHz outputs, respectively. The safe clock source can be scaled down by a configurable pre-scaler 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 switches 32 kHz clock from XOSC32K to the safe clock, the XOSC32K is still enabled. If required, 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.


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

**Note:** If an application can recover the XOSC32K from a Clock Failure Detection, the application can switch back to the XOSC32K clock (CFDCTRL.SWBACK = 1). Entering in Low-Power Sleep mode will not be possible if the clock switch back can not complete due to a persistent failure. Therefore, the user should disable the Clock Failure Detector (CFDCTRL.CFDEN = 0) until the clock switch back has completed (CFDCTRL.SWBACK = 0).

**Prescaler**

The CFD has an internal configurable prescaler to generate the safe clock from the OSCULP32K oscillator. The prescaler allows users to scale down the OSCULP32K oscillator clock to half so the safe clock frequency is not higher than the XOSC32K clock frequency monitored by the CFD.

The prescaler is applied on both outputs (32 kHz and 1 kHz) of the safe clock.

**Note:** For an external crystal oscillator at 32 kHz and the OSCULP32K frequency is 32 kHz, the XOSC32K.CFDPRESC must be set to 0 for a safe clock of equal frequency.

**Sleep Mode**

The CFD is halted depending on configuration of the XOSC32K and the peripheral clock request. In sleep modes, CFD operation is automatically disabled when the external oscillator is not requested to run by a peripheral. For additional information, refer to the *XOSC32K Sleep Mode Behavior* table. The CFD interrupt can be used to wakeup the device from sleep modes.

### 20.5.4. 32 kHz Ultra-Low Power Internal Oscillator (OSCULP32K) Operation

The OSCULP32K provides a low-speed and ultra-low power clock source. The OSCULP32K is a factory-calibrated oscillator under typical voltage and temperature conditions.

The OSCULP32K is enabled by default after a Power-on Reset (POR) and will always run except during POR.

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.

### 20.5.5. 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.

### 20.5.6. RTC Clock Selection Control

Before enabling the system RTC module, the RTC clock must be selected first. All outputs from this oscillator source are valid RTC clock sources. The RTC Clock Selection is available in the Selection Control register (CLKSELCTRL.RTCSEL). To ensure proper operation, it is highly recommended to disable the RTC module before changing its clock source.

### 20.5.7. Interrupts

The OSC32KCTRL has the following interrupt sources:

- **XOSC32KRDY – 32 kHz Crystal Oscillator Ready:** A 0-to-1 transition on the STATUS.XOSC32KRDY bit is detected and XOSC32K ready interrupt request or XOSC32K ready interrupt wake up request are set.
- **XOSC32KFAIL – Clock Failure Detector:** A 0-to-1 transition on the STATUS.XOSC32KFAIL bit is detected and XOSC32K clock fail detector interrupt request or XOSC32K clock fail detector interrupt wake up request are set.

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).


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

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 two interrupt request line each for XOSC32KRDY and XOSC32KFAIL interrupt sources.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

### 20.5.8. Events

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 and event line will go high. When the CFD is switched to the safe clock, the CFD clock failure will not be output on the Event Output.

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.XOSC32KSW) 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.

## 20.6. Peripheral Dependencies

**Table 20-1.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| OSC32KCTRL | 0x45008000 | 7 : XOSC32KRDY<br>8 : XOSC32KFAIL | **APB:**<br>CLKMSK[0]<br>MASK7 | --- | 7<br>STATUS0<br>PERID7 | | Generators:<br>**XOSC32KFAIL:**<br>CHANNEL[3] | PD_CORE_BU |


© 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 | 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 | CLKSELCTRL | 7:0 | | | | | | | RTCSEL[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | CFDCTRL | 7:0 | | | | | | CFDPRESC | SWBACK | CFDEN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | EVCTRL | 7:0 | | | | | | | | CFDEO |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | XOSC32K | 7:0 | ONDEMAND | | | | | XTALEN | ENABLE | |
| | | 15:8 | | | | | STARTUP[3:0] | | | |
| | | 23:16 | | | CGM[3:0] | | | | | |
| | | 31:24 | | | | | | | | |


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

### 20.7.1. 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).

**Table 20-2.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | | | | RW | | RW |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector Interrupt Enable
**Note:** Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Fail Interrupt Enable bit, (i.e. XOSC32KFAIL), which disables the XOSC32K Clock Failure interrupt
- Writing a one to the same corresponding bit in the INTENSET register

| Value | Description |
|---|---|
| 0 | The XOSC32K Clock Fail Detect Interrupt is disabled. |
| 1 | The XOSC32K Clock Fail Detect Interrupt 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
**Note:** Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit, (XOSC32KRDY), will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt


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

- Writing a one to the same corresponding bit in the INTENSET register

| 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.

### 20.7.2. 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).

**Table 20-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | | | | RW | | RW |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector Interrupt Enable
**Note:** Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Fail Interrupt Enable bit, (i.e. XOSC32KFAIL), which disables the XOSC32K Clock Failure interrupt
- Writing a one to the same corresponding bit in the INTENCLR register

| Value | Description |
|---|---|
| 0 | The XOSC32K Clock Fail Detect Interrupt is disabled. |
| 1 | The XOSC32K Clock Fail Detect Interrupt 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
**Note:** Writing a '0' to this bit has no effect.

This bit is cleared under the following conditions:

- Writing a '1' to this bit, (XOSC32KRDY), will clear the XOSC32K Ready Interrupt Enable bit, which disables the XOSC32K Ready interrupt


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

- Writing a one to the same corresponding bit in the INTENCLR register

| 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.

### 20.7.3. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** -

**Note:** Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

**Table 20-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | | | | RW | | RW |
| Reset | | | | | | 0 | | 0 |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector
**Note:** Writing a '0' to this bit has no effect.

This bit is set by hardware under the following conditions:

- This flag is set by hardware on a zero-to-one transition of the XOSC32K Clock Failure Detection bit in the (STATUS.XOSC32KFAIL) register and will generate an Interrupt request if INTENSET.XOSC32KFAIL is '1'

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Clock Failure Detection flag

| Value | Description |
|---|---|
| 0 | No XOSC32K 32.768kHz Clock Fail Detected. |
| 1 | XOSC32K 32.768kHz Clock Fail Detected. |

**Bit 0 – XOSC32KRDY** XOSC32K Ready
**Note:** Writing a '0' to this bit has no effect.

This bit is set by hardware under the following conditions:


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

- 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

This bit is cleared under the following conditions:

- Writing a '1' to this bit will clear the XOSC32K Ready interrupt flag

| 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.

### 20.7.4. Power and Clocks Status

**Name:** STATUS
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** -

**Table 20-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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

**Note:** This bit is set by hardware if a 32.768kHz clock fail detect occurs and clock fail detect is enabled, (i.e. CFDCTRL.CFDEN =1 and INTFLAG.XOSC32KFAIL=1 or STATUS.XOSC32KFAIL=1).

| Value | Description |
|---|---|
| 0 | XOSC32K is not switched and the clock source is provided by the external 32.768kHz oscillator source. |
| 1 | XOSC32K is switched to the internal OSCULP32K oscillator clock. Both 32.768kHz and 1.024kHz outputs of the XOSC32K are replaced by the respective OSCULP32K 32 kHz and 1 kHz outputs to the respective logic modules. |

**Bit 2 – XOSC32KFAIL** XOSC32K Clock Failure Detector

**Note:** This bit is set by hardware if a 32.768kHz clock fail detect occurs and clock fail detect is enabled, (i.e. CFDCTRL.CFDEN =1 and INTFLAG.XOSC32KFAIL=1).

| Value | Description |
|---|---|
| 0 | No XOSC32K 32.768kHz clock fail detection. |
| 1 | XOSC32K 32.768kHz external crystal/clock fail detect. Hardware clock switch will be initiated to OSCULP32K oscillator clocks. |

**Bit 0 – XOSC32KRDY** XOSC32K Ready

**Note:** This bit is set and cleared by hardware based on the status of the of the active XOSC32K 32.768kHz clock source.


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

| 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.

### 20.7.5. Clock Selection Control

**Name:** CLKSELCTRL
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 20-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | RTCSEL[1:0] | |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – RTCSEL[1:0]** RTC Clock Selection
These bits select the RTC clock source.

| Value | Name | Description |
|---|---|---|
| 0x0 | ULP32K | 32 kHz from 32 kHz internal ULP oscillator |
| 0x1 | ULP1K | 1 kHz from 32 kHz internal ULP oscillator |
| 0x2 | XOSC32K | 32.768 kHz from 32 kHz external crystal oscillator |
| 0x3 | XOSC1K | 1.024 kHz from 32 kHz external oscillator |

| Value | Name | Description |
|---|---|---|
| 0x0 | ULP32K | 32.768kHz from 32kHz internal ULP oscillator |
| 0x1 | ULP1K | 1.024kHz from 32kHz internal ULP oscillator |
| 0x2 | XOSC32K | 32.768kHz from 32.768kHz external crystal oscillator |
| 0x3 | XOSC1K | 1.024kHz from 32.768kHz internal oscillator |


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

### 20.7.6. Clock Failure Detector Control

**Name:** CFDCTRL
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 20-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CFDPRESC | SWBACK | CFDEN |
| Access | | | | | | RW | RW | RW |
| 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 controls the XOSC32K output switch back to the external clock or crystal oscillator 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.
**Note:** After setting CFDEN to enable clock failure detection, STATUS.XOSC32KFAIL will always be set. This first detection must be ignored. Subsequent setting of this bit will indicate actual clock failure events.


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

| Value | Description |
|---|---|
| 0 | The CFD is disabled. |
| 1 | The CFD is enabled. |


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

### 20.7.7. Event Control

**Name:** EVCTRL
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 20-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | CFDEO |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – CFDEO** Clock Failure Detector Event Output 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.
**Note:** To prevent false event generation, the CFDEO bit must be set or cleared *only* when the XOSC32K is Disabled, (i.e. XOSC32K.ENABLE=0).

| 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.

### 20.7.8. 32kHz External Crystal Oscillator (XOSC32K) Control

**Name:** XOSC32K
**Offset:** 0x1C
**Reset:** 0x00200080
**Property:** PAC Write-Protection

**Table 20-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | CGM[3:0] | | | | | |
| Access | | | RW | RW | RW | RW | | |
| Reset | | | 1 | 0 | 0 | 0 | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | STARTUP[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | | | | | XTALEN | ENABLE | |
| Access | RW | | | | | RW | RW | |
| Reset | 1 | | | | | 0 | 0 | |

**Bits 21:18 – CGM[3:0]** Control Gain Mode
These bits control the gain of the external crystal oscillator.
These bits are enable-protected.
**Note:**

1. These bits are enable-protected. They cannot be written to if XOSC32K.ENABLE = 1.

**Note:** For additional information regarding ESR, please consult the *Electrical Characteristics* chapter.

| Value | Name | Description |
|---|---|---|
| 0x0 | CGM0 | the lower Control Gain Mode value |
| 0x1 | CGM1 | a higher Control Gain Mode value than CGM0 |
| 0x2 | CGM2 | a higher Control Gain Mode value than CGM1 |
| 0x3 | CGM3 | a higher Control Gain Mode value than CGM2 |
| 0x4 | CGM4 | a higher Control Gain Mode value than CGM3 |
| 0x5 | CGM5 | a higher Control Gain Mode value than CGM4 |
| 0x6 | CGM6 | a higher Control Gain Mode value than CGM5 |
| 0x7 | CGM7 | a higher Control Gain Mode value than CGM6 |
| 0x8 | CGM8 | a higher Control Gain Mode value than CGM7 |
| 0x9 | CGM9 | a higher Control Gain Mode value than CGM8 |
| 0xA | CGM10 | a higher Control Gain Mode value than CGM9 |
| 0xB | CGM11 | a higher Control Gain Mode value than CGM10 |


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

| Value | Name | Description |
|---|---|---|
| 0xC | CGM12 | a higher Control Gain Mode value than CGM11 |
| 0xD | CGM13 | a higher Control Gain Mode value than CGM12 |
| 0xE | CGM14 | a higher Control Gain Mode value than CGM13 |
| 0xF | CGM15 | the highest Control Gain Mode value |

**Bits 11:8 – STARTUP[3:0]** Startup Mode

These bits select the start-up time for the XOSC32K oscillator. The OSCULP32K oscillator is used to clock the start-up counter. The given time assumes an XOSC32K crystal frequency of 32.768 kHz. After the XOSC32K.STARTUP time has expired, the XOSC32K clock is released internally after the selected programmable startup clock cycles plus 3 additional XOSC32 periods. The Clock Fail Detect (CFD) monitoring also starts when the clock is released for internal use. The user selected start-up time must equal or exceed the start-up time defined in the electrical characteristics.

| STARTUP[2:0] | OSCULP32K Clock Cycles | Plus | XOSC32K Clock Cycles | Equal | Approximate Equivalent Time |
|---|---|---|---|---|---|
| 0x0 | 1 | + | 3 | = | ~122 µs |
| 0x1 | 16 | + | 3 | = | ~580 µs |
| 0x2 | 32 | + | 3 | = | ~1.07 ms |
| 0x3 | 2048 | + | 3 | = | ~62.6 ms |
| 0x4 | 4096 | + | 3 | = | ~125.1 ms |
| 0x5 | 8192 | + | 3 | = | ~250.1 ms |
| 0x6 | 16384 | + | 3 | = | ~500.1 ms |
| 0x7 | 32768 | + | 3 | = | ~1s |
| 0x8 | 65536 | + | 3 | = | ~2s |
| 0x9 | 131072 | + | 3 | = | ~4s |
| 0xA | 262144 | + | 3 | = | ~8s |
| 0xB – 0xF | --- | --- | --- | --- | Reserved |

**Note:** These bits are enable-protected. They cannot be written to if XOSC32K.ENABLE = 1.
These bits are valid only when XOSC32K.XTALEN = 1, crystal XOSC32K selected.

| Value | Name | Description |
|---|---|---|
| 0x0 | 1CYCLE | 1 ULP clock cycle |
| 0x1 | 16CYCLES | 16 ULP clock cycles |
| 0x2 | 32CYCLES | 32 ULP clock cycles |
| 0x3 | 2048CYCLES | 2048 ULP clock cycles |
| 0x4 | 4096CYCLES | 4096 ULP clock cycles |
| 0x5 | 8192CYCLES | 8192 ULP clock cycles |
| 0x6 | 16384CYCLES | 16384 ULP clock cycles |
| 0x7 | 32768CYCLES | 32768 ULP clock cycles |
| 0x8 | 65536CYCLES | 65536 ULP clock cycles |
| 0x9 | 131072CYCLES | 131072 ULP clock cycles |
| 0xA | 262144CYCLES | 262144 ULP clock cycles |

**Bit 7 – ONDEMAND** On Demand Mode

This bit controls how the XOSC32K behaves when a peripheral clock request is detected.
**Note:** This bit is enable-protected. It cannot be written to if XOSC32K.ENABLE = 1.

| Value | Description |
|---|---|
| 0 | XOSC32K always run |
| 1 | Only run if requested by a peripheral |

**Bit 2 – XTALEN** Crystal Oscillator Enable

This bit controls the connections between the I/O pads and the external clock or crystal oscillator.
**Note:** This bit is enable-protected. It cannot be written to if XOSC32K.ENABLE = 1.


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

| 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

**Note:** It is necessary to wait for STATUS.XOSC32KRDY = 0 before enabling the XOSC32K.

| Value | Description |
|---|---|
| 0 | The 32K oscillator is disabled. |
| 1 | The 32K oscillator is enabled. |


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

# 21. Watchdog Timer (WDT)

## 21.1. 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 cause 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, this will cause a system reset. Compared to the normal mode, this can also catch situations where a code error causes the WDT to be cleared frequently.

When enabled, the WDT runs in active mode and selectively runs in Standby Sleep mode. It is asynchronous and runs from a CPU-independent internal clock source. The WDT continues operation and causes a system reset even if the main clocks fail. Even though there can be time between the reset condition occurring and the system reset asserting, it cannot be prevented. This includes but is not limited to a software attempt to disable the WDT.

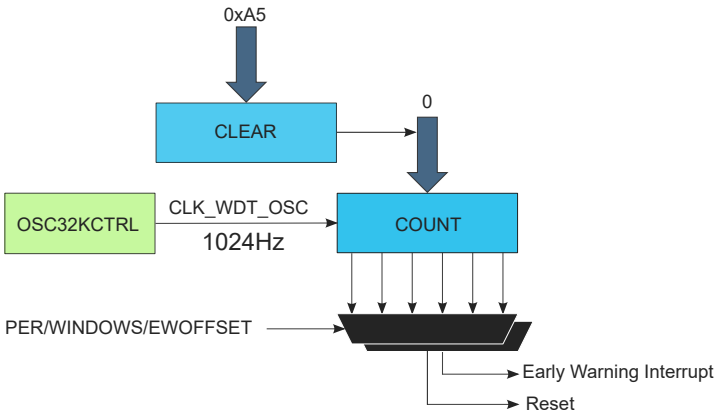
## 21.2. Features

The following are key features of the WDT module:

- 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 mode
  - 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

## 21.3. Block Diagram

**Figure 21-1.** WDT Block Diagram





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

## 21.4. Functional Description

### 21.4.1. 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 21-1.** 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 |

### 21.4.2. Basic Operation

#### 21.4.2.1. Initialization

The WDT has enable-protected registers which 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 Period bits in the Configuration register (CONFIG.WINDOW) must be defined. The Window Enable bit (CTRLB.WEN=1) selects window mode.

When enabling the WDT from the disabled state, the WDT count starts from 0.

#### 21.4.2.2. Configurable Reset Values

After a Power-on Reset, some registers will be loaded with initial values from the *USER CFG page*.

This includes the following bits and bit groups:

- Enable bit in the Control A register, CTRLA.ENABLE
- Always-On bit in the Control A register, CTRLA.ALWAYSON
- Run In Standby Enable bit in the Control A register (CTRLA.RUNSTDBY)
- 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

#### 21.4.2.3. 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'.

#### 21.4.2.4. 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


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

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.

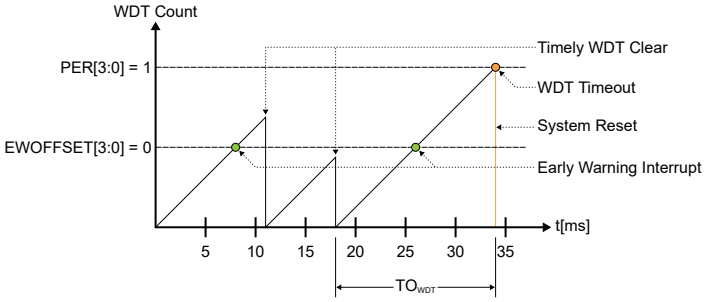
The WDT is cleared and a new WDT time-out period is started by writing 0xA5 to the Clear register (CLEAR). Writing any other value than 0xA5 to CLEAR will issue an immediate system reset.

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 21-2.** Normal-Mode Operation




### 21.4.2.5. 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.

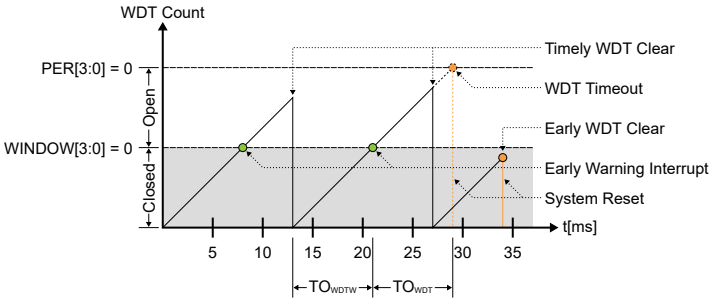
The closed window period is defined by the Window Period bits in the Configuration register (CONFIG.WINDOW), and the open window period is defined by the Period bits in the Configuration register (CONFIG.PER).

By default, the Early Warning interrupt is disabled. If it is desired, the Early Warning Interrupt Enable bit in the Interrupt Enable SET 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.


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

**Figure 21-3.** Window-Mode Operation




### 21.4.3. Clocks

The WDT bus clock (CLK_WDT_APB) can be enabled and disabled (masked) in the *Main Clock module (MCLK)*.

A 1024 Hz oscillator clock (CLK_WDT_OSC) is required to clock the WDT internal counter.

The CLK_WDT_OSC clock is sourced from the clock of the internal Ultra Low-Power Oscillator (OSCULP32K).

**CAUTION** Watchdog time-out period variations must be considered when implementing software that uses the WDT to ensure that the time-out periods used are valid for all devices. Refer to the *OSCULP32K Electrical Specifications section* of the Electrical Characteristics chapter.

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* section for further details.

**Note:** The FREQM module can be used to calibrate the output of the OSCULP32K clock against a refererence clock, thereby reducing the uncertainty in setting WDT windows.

### 21.4.4. 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 at the system level to generate one combined interrupt request to the *Nested vector Interrupt Controller*. 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.


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

### 21.4.5. Sleep Mode Operation

The Run-In-Standby bit in Control A (CTRLA.RUNSTDBY) control the behavior of the WDT during Standby Sleep mode. When the bit is zero, the watchdog is disabled during sleep, but maintains its current configuration. When CTRLA.RUNSTDBY is '1', the WDT continues to operate during sleep.

### 21.4.6. Debug Operation

When the CPU is halted in debug mode the WDT will halt normal operation.

### 21.4.7. Synchronization

Some registers (or bit fields within a register) require synchronization when read and/or written.

Synchronization is denoted by the "Read-Synchronized" (or "Read-Synchronized Bits”) and/or "Write-Synchronized" (or "Write-Synchronized Bits”) property in each individual register description.

For more details, refer to *Register Synchronization*.

### 21.4.8. Additional Features

#### 21.4.8.1. 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. Therefore, 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.

The following able shows the operation of the WDT for CTRLA.ALWAYSON = 1.

**Table 21-2.** 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 |

#### 21.4.8.2. 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


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

wake up and clear the Watchdog Timer, after which the system can perform other tasks or return to sleep mode.

> 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.

## 21.5. Peripheral Dependencies

**Table 21-3.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| WDT | 0x4400E000 | 11 : EW_A | **APB:** CLKMSK[0] MASK11 | --- | 11 STATUS0 PERID11 | | | PD_CORE_SW |


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

## 21.6. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ALWAYSON | RUNSTDBY | | | | 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 | RUNSTDBY | WEN | ENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | CLEAR | 7:0 | CLEAR[7:0] | | | | | | | |


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

### 21.6.1. Control

**Name:** CTRLA
**Offset:** 0x0
**Reset:** 0x00
**Property:** RW

**Table 21-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ALWAYSON | RUNSTDBY | | | | WEN | ENABLE | |
| Access | RW | RW | | | | RW | RW | |
| Reset | 0 | 0 | | | | 0 | 0 | |

**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.
These bit is loaded from User Configuration FUCFG0 at startup.
**Note:** This bit is not enable-protected.

| 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 6 – RUNSTDBY** Run During Standby
This bit controls the behavior of the watchdog during Standby Sleep mode.

- When CTRLA.ALWAYSON = 0, this bit is enable-protected by CTRLA.ENABLE.
- When CTRLA.ALWAYSON = 1, this bit is not enable-protected by CTRLA.ENABLE.

These bit is loaded from User Configuration FUCFG0 at start up.

| Value | Description |
|---|---|
| 0 | The WDT is disabled during Standby sleep mode. |
| 1 | The WDT is enabled continues to operate during Standby sleep mode. |

**Bit 2 – WEN** Watchdog Timer Window Mode Enable
This bit enables Window mode.

- When CTRLA.ALWAYSON = 0, this bit is enable-protected by CTRLA.ENABLE.
- When CTRLA.ALWAYSON = 1, this bit is not enable-protected by CTRLA.ENABLE.

These bit is loaded from User Configuration FUCFG0 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.


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

These bit is loaded from User Configuration FUCFG0 at startup.
**Note:** This bit is write-synchronized: SYNCBUSY.ENABLE must be checked to ensure the CTRLA.ENABLE synchronization is complete.

**Note:** This bit is not enable-protected.

| Value | Description |
|---|---|
| 0 | The WDT is disabled. |
| 1 | The WDT is enabled. |


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

### 21.6.2. Configuration

**Name:** CONFIG
**Offset:** 0x1
**Reset:** 0xBB
**Property:** RW

**Table 21-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | WINDOW[3:0] | | | | PER[3:0] | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |

**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 1024 Hz CLK_WDT_OSC clock. These bits are loaded from *User Configuration FUCFG0* 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 |

**Bits 3:0 – PER[3:0]** Time-Out Period
These bits determine the watchdog time-out period as a number of 1024 Hz CLK_WDTOSC clock cycles. In Window mode operation, these bits define the open window period. These bits are loaded from *User Configuration FUCFG0* 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 |


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

### 21.6.3. Early Warning Interrupt Control

**Name:** EWCTRL
**Offset:** 0x2
**Reset:** 0x0B
**Property:** RW

**Table 21-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bits 3:0 – EWOFFSET[3:0]** Early Warning Interrupt Time Offset
These bits determine the number of GCLK_WDT clock cycles between the start of the watchdog time-out period and the generation of the Early Warning interrupt. These bits are loaded from *User Configuration FUCFG0* 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 |


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

### 21.6.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x4
**Reset:** 0x00
**Property:** RW

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.

**Table 21-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | EW |
| Access | | | | | | | | RW |
| 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.

### 21.6.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x5
**Reset:** 0x00
**Property:** RW

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.

**Table 21-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | EW |
| Access | | | | | | | | RW |
| 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.

### 21.6.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x6
**Reset:** 0x00
**Property:** RW

**Note:** Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

**Table 21-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | EW |
| Access | | | | | | | | RW |
| 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.

### 21.6.7. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x8
**Reset:** 0x00000000
**Property:** R

**Table 21-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | RUNSTDBY | WEN | ENABLE | |
| Access | | | R | R | R | R | R | |
| Reset | | | 0 | 0 | 0 | 0 | 0 | |

**Bit 5 – 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 4 – 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 3 – RUNSTDBY** Run During Standby Synchronization Busy

| Value | Description |
|---|---|
| 0 | Write synchronization of the CTRLA.RUNSTDBY bit is complete. |
| 1 | Write synchronization of the CTRLA.RUNSTDBY bit is ongoing. |

**Bit 2 – WEN** Window Enable Synchronization Busy

| Value | Description |
|---|---|
| 0 | Write synchronization of the CTRLA.WEN bit is complete. |


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

| Value | Description |
|---|---|
| 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.

### 21.6.8. Clear

**Name:** CLEAR
**Offset:** 0xC
**Reset:** 0x00
**Property:** W

**Note:** This register is write-synchronized: SYNCBUSY.CLEAR must be checked to ensure the CLEAR register synchronization is complete.

**Table 21-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.
**Note:** This bit field is write-synchronized: SYNCBUSY.CLEAR must be checked to ensure the CLEAR.CLEAR synchronization is complete.

| Value | Name | Description |
|---|---|---|
| 0xA5 | KEY | Clear Key |


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

# 22. Frequency Meter (FREQM)

## 22.1. Overview

The Frequency Meter (FREQM) can be used to accurately measure the frequency of a clock source by comparing it to a known reference clock. It 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*GCLK_FREQM_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) over which the duration of a measurement is to be done.

REFNUM is the number of Reference clock cycles selected in the Configuration A register (CFGA.REFNUM) over which the duration of a measurement is to be done.

The frequency of the measured clock, $f_{CLK_MSR}$, is calculated by:

$$f_{GCLK_FREQM_MSR} = \left(\frac{VALUE + 1 + Error}{REFNUM + 1}\right) f_{GCLK_FREQM_REF}$$

Where, *Error* represents the error introduced by the synchronization mechanism. The error can be a maximum of two measured clock cycles.

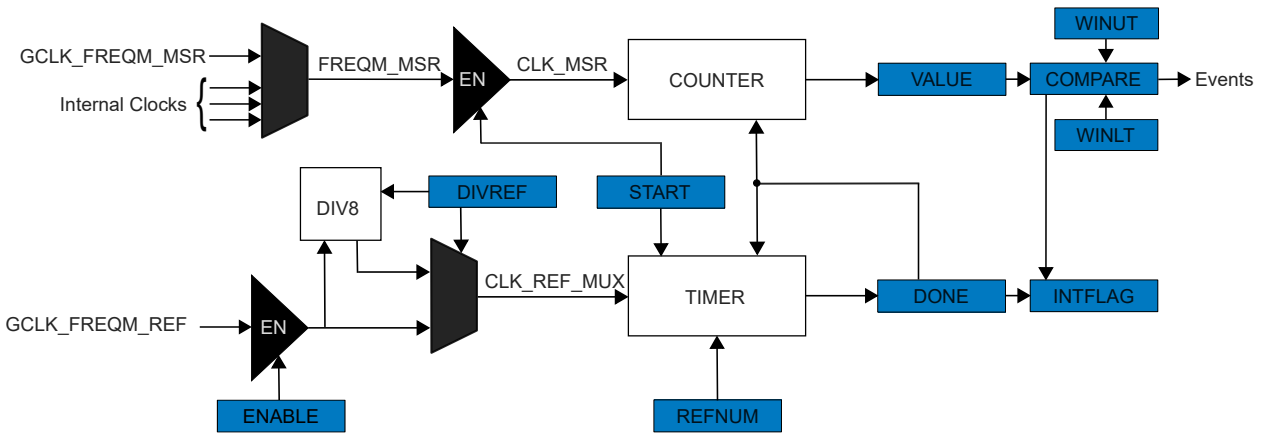
## 22.2. Features

The following are key features of the FREQM module:

- Ratio can be measured with 24-bit accuracy
- Accurately measures the frequency of an input clock source from GCLK_FREQM_MSR 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
- Free-running support enabling continuous clock monitor with flexible window threshold selection

## 22.3. Block Diagram

**Figure 22-1.** FREQM Block Diagram




## 22.4. 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.

### 22.4.1. 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.

### 22.4.2. Power Management

The FREQM will continue to operate in any sleep mode where the selected source clock is running. The FREQM'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. Refer to the Power Manager chapter for details on the different sleep modes..

### 22.4.3. Clocks

The clock for the FREQM bus interface (CLK_APB_FREQM) is enabled and disabled by the Main Clock Controller, Refer to "Peripheral Clock Masking" for details and default status of this clock.

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.

The FREQM has also the capability to use other internal clocks as source for frequency measurement. These clocks are chip specific and must be configured and enabled before using the FREQM.

### 22.4.4. Interrupts

The interrupt request line is connected to the interrupt controller. Using FREQM interrupt requires the interrupt controller to be configured first.

### 22.4.5. Debugging 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.

### 22.4.6. 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.

## 22.5. Functional Description

### 22.5.1. Basic Operation

#### 22.5.1.1. Initialization

Before enabling FREQM, the device and peripheral must be configured:


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

- The generic reference clock (GCLK_FREQM_REF) must be configured and enabled.
- The Measurement Clock Selection bits in Configuration A register (CFGA.MSRSEL) must be configured and the selected clock 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).

The following registers are enable-protected, meaning that they can only be written when the FREQM is disabled (CTRLA.ENABLE = 0):

- Configuration A register (CFGA)
- Control C register (CTRLC )
- Event Control register (EVCTRL )
- Window Monitor Lower Threshold register (WINLT )
- Window Monitor Upper Threshold register (WINUT )

Enable-protection is denoted by the "Enable-Protected" property in the register description.

The following register bits are enable-protected, meaning that they can only be written when the FREQM is disabled (CTRLA.ENABLE = 0):

- On Demand Control bit in Control A register (CTRLA.ONDEMAND )
- Run During Standby bit in Control A register (CTRLA.RUNSTDBY)
- Free Running Mode bit in Control A register (CTRLA.FREERUN )

The 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'.

#### 22.5.1.2. 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.

#### 22.5.1.3. 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.

**Notes:**

1. The REFNUM field must be written before the FREQM is enabled.
2. The measurement starts when both reference and measurement clocks are running. If the GCLK_FREQM_MSR clock is stopped, the measurement halts. When the clock is running again, a new measurement will start when requested.

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 the Status register (STATUS.BUSY) is set when the measurement starts, and cleared when the measurement is complete.

**Note:** Writing a '1' to the START bit in the Control B register during a measurement will be ignored.


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

The Measurement Done interrupt provides an alternative way to detect measurement completion. 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:

$$f_{CLK_MSR} = \left(\frac{VALUE + 1 + Error}{REFNUM + 1}\right) f_{GCLK_FREQM_REF}$$

Where,

*Error* represents the error introduced by the synchronization mechanism. The error can be a maximum of two CLK_MSR cycles.

**Notes:**

1. To make sure the measurement result (VALUE.VALUE[23:0]) is valid, the overflow status (STATUS.OVF) must be checked.
2. Due to asynchronous operation, the VALUE Error measurement can be up to two samples.

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.

#### 22.5.1.4. Window Monitor Mode

Window Monitor Mode causes the FREQM to compare each measurement result to pre-configured thresholds. It can compare the value to see if it is below a lower, above an upper, within the upper and lower, or outside of the upper and lower thresholds. The Window Monitor Mode is selected by setting the Window Mode bits in the Control C register (CTRLC.WINMODE). Threshold values must be written in the Window Monitor Lower Threshold register (WINLT.WINLT) and Window Monitor Upper Threshold register (WINUT.WINUT).

The FREQM also supports an interrupt request for Window Monitor operation: When the Window Monitor bit in Interrupt Enable Set register (INTENSET.WINMON) is '1' and a measurement meets the criteria defined in the Window Mode bits in the Control C register (CTRLC.WINMODE), the Window Monitor bit in the Interrupt Flag Status and Clear register (INTFLAG.WINMON) will be set and an interrupt request is generated.

#### 22.5.1.5. Free Running Mode

Free Running Mode causes the FREQM to automatically start a new measurement after each measurement is complete. The Free Running Mode can be disabled by disabling the module, or by setting the Software Reset bit in the Control A register (CTRLA.SWRST = 1).

**Important:** In free running mode, when a new measurement starts, the VALUE register will follow the MSR counter value. It is not recommended to read the VALUE register when the free running operating mode is enabled.

**Notes:**

1. In free running mode, the event detection is disabled. As consequence, all incoming events are ignored and no actions will be triggered inside the FREQM module.
2. Writing to CTRLB.START when a measurement is in progress is ignored. The bit is always read zero.

### 22.5.2. Interrupts

The FREQM has two interrupt sources:


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

- DONE: A frequency measurement is done.
- WINMON: A window compare valid condition is detected.

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 the Interrupt Flag Status and Clear (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 Interrupt Flag Status and Clear (INTFLAG) register to determine which interrupt condition is present.

This interrupt is a synchronous wake-up source.

**Note:** Interrupts must be globally enabled for interrupt requests to be generated.

### 22.5.3. Events

The FREQM can generate the following output events:

- Done (DONE): Measurement Done.
- Window Monitor (WINMON): Generated when the window monitor condition matches.

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 FREQM can take the following actions on an input event:

- Start measurement (START): Start a new measurement. If a new START event is detected during an ongoing measurement, the event is ignored.

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.

By default, the FREQM will detect a rising edge on the incoming event. If the FREQM 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. STARTINV).

### 22.5.4. Sleep Mode Operation

During sleep modes, the ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) controls the behavior of the FREQM clock requests, in cases where the FREQM is enabled (CTRLA.ENABLE = 1) during a Sleep mode.

**Table 22-1.** FREQM Sleep Behavior

| CTRLA.RUNSTDBY | CTRLA.ONDEMAND | CTRLA.ENABLE | Description |
|---|---|---|---|
| X | x | 0 | Disabled |
| 0 | 0 | 1 | Always request the clocks in all sleep modes, except in Standby mode. |
| 0 | 1 | 1 | On-demand clocks request in all sleep modes, except in Standby mode. |
| 1 | 0 | 1 | Always request the clocks in all sleep modes. |
| 1 | 1 | 1 | On-demand clocks request in all sleep modes. |


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

**Note:** By default, if RUNSTDBY = 0, an ongoing measurement is stopped when the device is going to Standby Sleep mode. A new measurement will be restarted when the device exits the Standby Sleep mode.

### 22.5.5. 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.

## 22.6. Peripheral Dependencies

**Table 22-2.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| FREQM | 0x4400C000 | 10 : DONE/ WINMON | **APB:** CLKMSK[0] MASK10 | **MSR0:** PCHCTRL[2] **MSR1:** PCHCTRL[3] **REF:** PCHCTRL[4] | 10 STATUS0 PERID10 | | Generators: **DONE:** CHANNEL[4] **WINMON:** CHANNEL[5] Users: **START:** USER[0],Resynch | PD_CORE_SW |


© 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 | CTRLA | 7:0 | ONDEMAND | RUNSTDBY | | | | FREERUN | ENABLE | SWRST |
| 0x01 | CTRLB | 7:0 | | | | | | | | START |
| 0x02 | CFGA | 7:0 | REFNUM[7:0] | | | | | | | |
| | | 15:8 | DIVREF | | | | | MSRSEL[2:0] | | |
| 0x04 | CTRLC | 7:0 | | | | | | WINMODE[2:0] | | |
| 0x05 | Reserved | | | | | | | | | |
| 0x06 | EVCTRL | 7:0 | | | WINMONEO | DONEEO | | | STARTINV | STARTEI |
| 0x07 | Reserved | | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | | | | | | | WINMON | DONE |
| 0x09 | INTENSET | 7:0 | | | | | | | WINMON | DONE |
| 0x0A | INTFLAG | 7:0 | | | | | | | WINMON | 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 | | | | | | | | |
| 0x14 ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | WINLT | 7:0 | WINLT[7:0] | | | | | | | |
| | | 15:8 | WINLT[15:8] | | | | | | | |
| | | 23:16 | WINLT[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | WINUT | 7:0 | WINUT[7:0] | | | | | | | |
| | | 15:8 | WINUT[15:8] | | | | | | | |
| | | 23:16 | WINUT[23:16] | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 22.7.1. Control A Register

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x80
**Property:** PAC Write-Protection

**Table 22-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | | | FREERUN | ENABLE | SWRST |
| Access | R/W | R/W | | | | R/W | R/W | R/W |
| Reset | 1 | 0 | | | | 0 | 0 | 0 |

**Bit 7 – ONDEMAND** On Demand Control
The On Demand operation mode allows the FREQM to request the clocks, depending on other peripheral requests. In On Demand operation mode, i.e., if the ONDEMAND bit has been previously set, the FREQM clocks will only be running when a measurement is requested by a peripheral or if free running operation mode is enabled. If there is no peripheral requesting the FREQM measurement, the module will not request any clock.
This bit is enable-protected.
**Note:** The ONDEMAND bit should always be set to '0'.

| Value | Description |
|---|---|
| 0 | The FREQM is always requesting the clocks when CTRLA.ENABLE is set. |
| 1 | The FREQM is requesting the clocks only when a peripheral is requesting a measurement, or if free running mode is enabled. |

**Bit 6 – RUNSTDBY** Run In Standby
This bit controls how the FREQM clocks will be requested during standby sleep mode.
This bit is enable-protected.

| Value | Description |
|---|---|
| 0 | The FREQM clocks are not requested during standby sleep mode. If CTRLA.ONDEMAND=1, the FREQM will request the clocks only if a measurement is requested. If CTRLA.ONDEMAND=0, the FREQM clocks will not be requested in standby sleep mode. |
| 1 | The FREQM clocks are requested in standby sleep mode. If CTRLA.ONDEMAND=1, the FREQM will request the clocks only if a measurement is requested. If CTRLA.ONDEMAND=0, the FREQM clocks will always be requested in standby sleep mode. |

**Bit 2 – FREERUN** Free Running Mode
This bit controls the free running operation mode.
This bit is enable-protected.

| Value | Description |
|---|---|
| 0 | The free running operating mode is disabled. |
| 1 | The free running operating mode is enabled. |

**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


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

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.

### 22.7.2. Control B Register

**Name:** CTRLB
**Offset:** 0x01
**Reset:** 0x00
**Property:** -

**Table 22-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 single measurement. |


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

### 22.7.3. Config A Register

**Name:** CFGA
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-protected

**Table 22-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DIVREF | | | | | MSRSEL[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 15 – DIVREF** Divide Reference Clock
Divides the reference clock by 8

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV1 | The reference clock is divided by 1 |
| 0x1 | DIV8 | The reference clock is divided by 8 |

**Bits 10:8 – MSRSEL[2:0]** Measurement Clock Selection

| Value | Name | Description |
|---|---|---|
| 0x0 | GCLK | GCLK Input Clock |
| 0x1 | GCLK0 | CPU Input Clock |

**Bits 7:0 – REFNUM[7:0]** Number of Reference Clock Cycles
Selects the duration of a measurement in number of CLK_FREQM_REF cycles.


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

### 22.7.4. Control C Register

**Name:** CTRLC
**Offset:** 0x04
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

**Table 22-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bits 2:0 – 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 | VALUE > WINLT |
| 0x2 | MODE2 | VALUE < WINUT |
| 0x3 | MODE3 | WINLT < VALUE < WINUT |
| 0x4 | MODE4 | !(WINLT < VALUE < WINUT) |
| 0x5 - 0x7 | | Reserved |


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

### 22.7.5. Event Control Register

**Name:** EVCTRL
**Offset:** 0x06
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

**Table 22-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | WINMONEO | DONEEO | | | STARTINV | STARTEI |
| Access | | | R/W | R/W | | | R/W | R/W |
| Reset | | | 0 | 0 | | | 0 | 0 |

**Bit 5 – WINMONEO** Window Monitor Event Out

| Value | Description |
|---|---|
| 0 | Window Monitor event output is disabled and an event will not be generated. |
| 1 | Window Monitor event output is disabled and an event will be generated. |

**Bit 4 – DONEEO** Measurement Done Event Out

| Value | Description |
|---|---|
| 0 | Measurement Done event output is disabled and an event will not be generated. |
| 1 | Measurement Done event output is disabled and an event will be generated. |

**Bit 1 – STARTINV** Start Measurement Event Invert Enable

| Value | Description |
|---|---|
| 0 | Start event input source is not inverted. |
| 1 | Start event input source is inverted. |

**Bit 0 – STARTEI** Start Measurement Event Input Enable

| Value | Description |
|---|---|
| 0 | A new measurement will not be triggered on any incoming event. |
| 1 | A new measurement will be triggered on any incoming event. |


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

### 22.7.6. Interrupt Enable Clear Register

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 22-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 1 – 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 Window Monitor interrupt.

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

**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.

### 22.7.7. Interrupt Enable Set Register

**Name:** INTENSET
**Offset:** 0x09
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 22-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 1 – 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 Enable bit, which enables the Window Monitor interrupt.

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

**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.

### 22.7.8. Interrupt Flag Register

**Name:** INTFLAG
**Offset:** 0x0A
**Reset:** 0x00
**Property:** -

**Note:** Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

**Table 22-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 1 – WINMON** Window Monitor
This flag is cleared by writing a '1' to it.
This flag is set on the next clock 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 will clear the WINMON interrupt flag.

**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.

### 22.7.9. Status Register

**Name:** STATUS
**Offset:** 0x0B
**Reset:** 0x00
**Property:** RW

**Table 22-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.<br>This bit is set only when the measurements are triggered by a software start (CTRLB.START = 1). |


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

### 22.7.10. Synchronization Busy Register

**Name:** SYNCBUSY
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** R

**Table 22-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 22.7.11. Count Value Register

**Name:** VALUE
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** R

**Table 22-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.
**Note:** The register value is refreshed each time a start event is detected, and as long as a measurement is ongoing. It is not recommended to read the register if the start of measurements are triggered by peripheral events or it is in Free Running Mode.


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

### 22.7.12. Window Monitor Lower Threshold

**Name:** WINLT
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 22-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | WINLT[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 23:0 – WINLT[23:0]** 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.

### 22.7.13. Window Monitor Upper Threshold

**Name:** WINUT
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 22-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | WINUT[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 23:0 – WINUT[23:0]** 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.

# 23. Real-Time Counter (RTC)

## 23.1. 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, 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 peripheral 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 32.768 kHz clock source, the minimum counter tick interval is 30.5 µs 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.

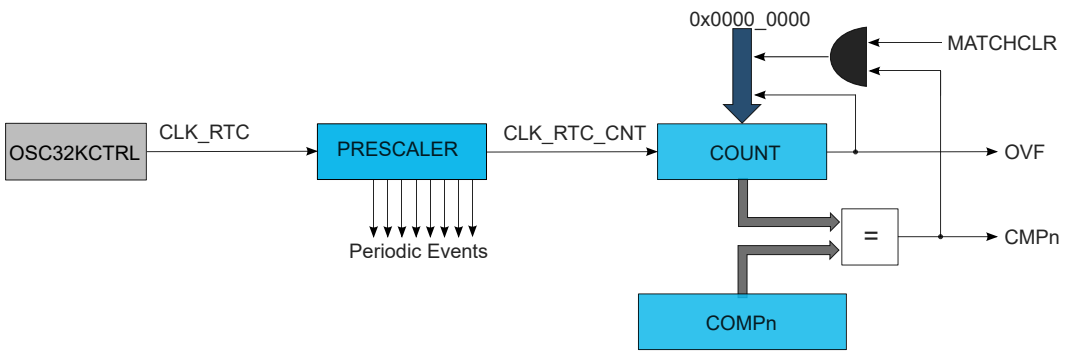
## 23.2. Features

The following are key features of the RTC module:

- 32-bit counter with 10-bit prescaler
- Multiple clock sources
- 32-bit or 16-bit counter mode
- 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
- 4 general purpose registers
- Tamper Detection
  - Timestamp on event or up to 8 inputs with debouncing
  - Active layer protection

## 23.3. Block Diagram

**Figure 23-1.** RTC Block Diagram (Mode 0 — 32-Bit Counter)





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

**Figure 23-2.** RTC Block Diagram (Mode 1 — 16-Bit Counter)

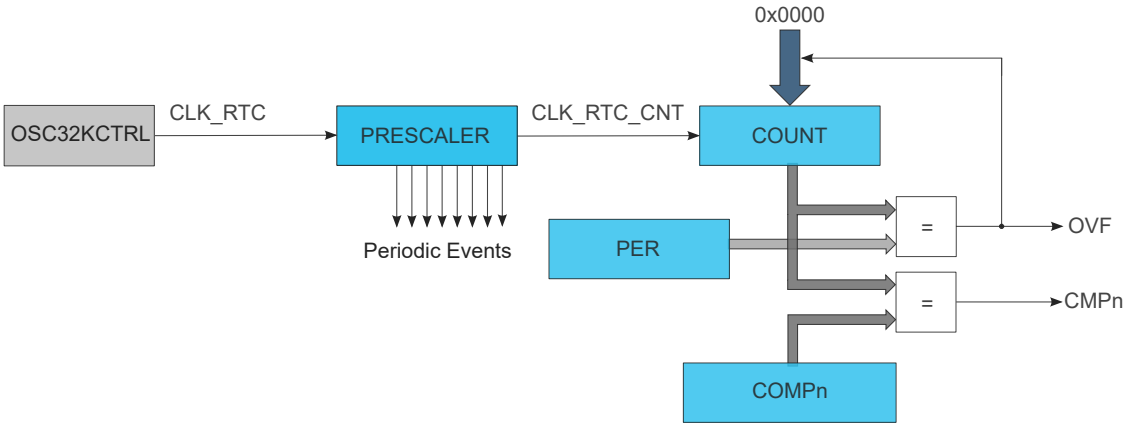



**Figure 23-3.** RTC Block Diagram (Mode 2 — Clock/Calendar)

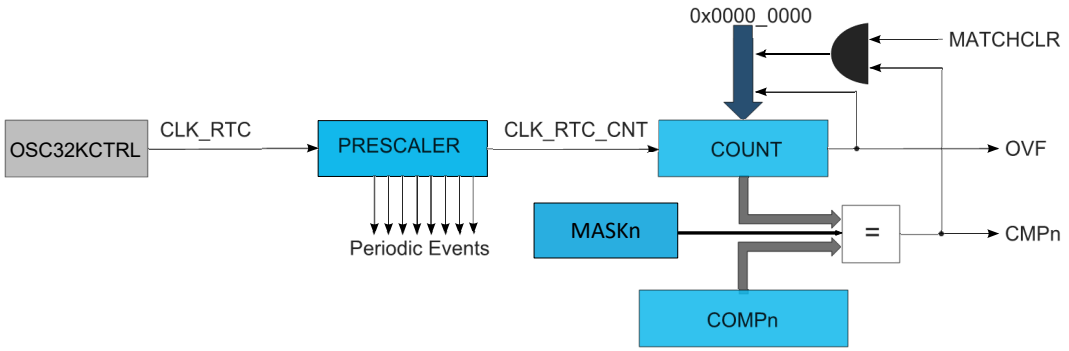
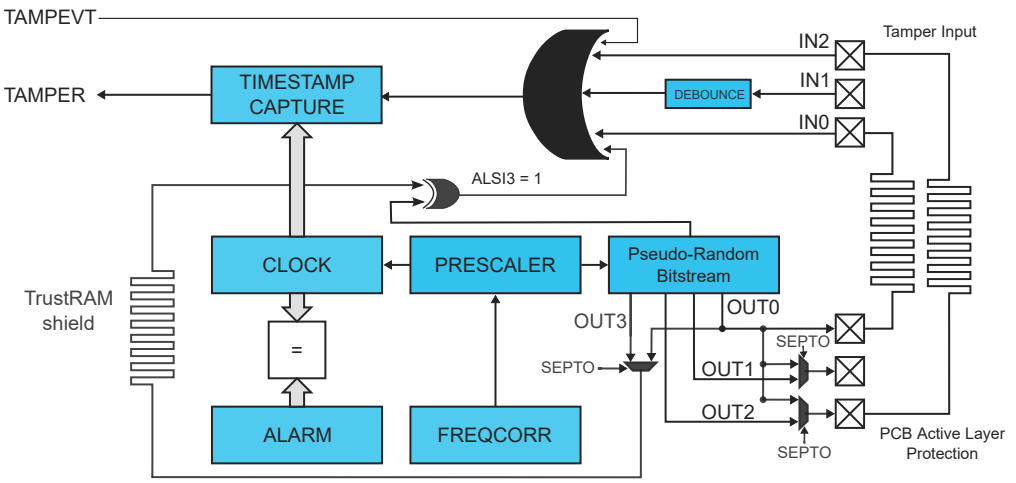



**Figure 23-4.** RTC Block Diagram (Tamper Detection Use Case)





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

## 23.4. Signal Description

**Table 23-1.** Signal Description

| Signal | Description | Type |
|---|---|---|
| RTC_IN[N-1:0] | Tamper Detection Input | Digital input |
| RTC_OUT[N-1:0] | Tamper Detection Output | Digital output |

**Note:** For this device, the number of tampers, N=8.

## 23.5. Peripheral Dependencies

| | |
|---|---|
| **Peripheral Name** | RTC |
| **Base Address** | 0X4407 2000 (Peripheral Bus A) |
| **NVIC IRQ Index:Source** | 15 : TAMPER 16 : Overflow (OVF) 17 : Period x (PERx), x=0,1,...7 18: Compare x (CMPx), x = 0,1,2,3 |
| **MCLK AXI/APB Clocks Index:Name** (1) | MCLK.CLKMSK0[15] |
| **GCLK Peripheral Channel Index:Clock Name** | NA |
| **PAC Peripheral Peripheral Identifier (PAC.WRCTRL)** | 12 |
| **APB Mask Register[Index]** | INTFLAGA[12] |
| **AHB Mask Register[Index]** | NA |
| **DMA Trigger Index:Source (DMAC.CHCTRLBk)** | 4 : RTC Timestamp (TIMESTAMP) |
| **EVSYS Users (EVSYS.USERm)** (2) | 1 : TAMPER (A) |
| **EVSYS Generators (EVSYS.CHANNELn)** | 5-12 : Periodic Interval n (RTC_PERx), x=0,1...7 13-16 : Compare x (RTC_CMPx), x = 0..3 17 : Tamper (RTC_TAMPER) 18 : Overflow (RTC_OVF) 19 : Daily Period (RTC_PERD) |
| **Power Domain** | VDDREG |

**Notes:**

1. Register Field: MCLK.CLKMSK{index/32}.MASK[index mod 32].
2. (A,S,R): A = Asynchronous path, S = Synchronous path, R = Resynchronized path.

### 23.5.1. 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 kHz or 1 kHz oscillator clock (CLK_RTC_OSC) is required to clock the RTC. This clock must be configured and enabled in the 32 kHz oscillator controller (OSC32KCTRL) 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 additional information.

### 23.5.2. 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 additional information.

### 23.5.3. 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.

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.


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

### 23.5.4. 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* for details on recommended crystal characteristics and load capacitors.

## 23.6. Functional Description

### 23.6.1. 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

### 23.6.2. Basic Operation

#### 23.6.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when the RTC is disabled (CTRLA.ENABLE = 0):

- The Operating Mode bits in the Control A register (CTRLA.MODE)
- The Clear on Match bit in the Control A register (CTRLA.MATCHCLR)
- The Prescaler bits in the Control A register (CTRLA.PRESCALER)
- The GP Registers Reset On Tamper Enable bit in the Control A register (CTRLA.GPTRST)
- The Clock Representation bit in the Control A register (CTRLA.CLKREP)

The following registers are enable-protected:

- The Control B (CTRLB) register
- The Event Control (EVCTRL) register
- The Tamper Control (TAMPCTRL) register
- The Tamper Control B (TAMPCTRLB) register

The 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. The 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.

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 Hz clock to the counter for correct operation.

The frequency of the RTC clock (CLK_RTC_CNT) is given by the following formula:

$CLK_RTC_CNT = [CLK_RTC / 2^{PRESCALAR}]$

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.


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

#### 23.6.2.2. 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.

#### 23.6.2.3. 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 the Block Diagram. 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..N-1, where N = 2). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..N-1, where N = 2) 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.

#### 23.6.2.4. 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 the Block Diagram. 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..N-1, where N = 4). When a compare match occurs, the Compare n Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.CMPn, n=0..N-1, where N = 4) is set on the next 0-to-1 transition of CLK_RTC_CNT.

#### 23.6.2.5. 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 the Block Diagram. 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
- Minutes
- Hours


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

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..N-1, where N = 2). When an alarm match occurs, the Alarm n Interrupt flag in the Interrupt Flag Status and Clear registers (INTFLAG.ALARMn) is set on the next 0-to-1 transition of CLK_RTC_CNT. E.g. For a 1Hz clock counter, it means the Alarm n 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, n=0..N-1, where N = 2). 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 (CTRLB.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 CTRLB.MATCHCLR is 1, INTFLAG.ALARMn and INTFLAG.OVF will both be set simultaneously on an alarm match with ALARMn.

### 23.6.3. 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.

### 23.6.4. 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 (INTENSET = 1), and disabled by setting the corresponding bit in the Interrupt Enable Clear register (INTENCLR = 1).


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

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 or more 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.

### 23.6.5. 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 (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.
- **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 '1' to an Event Input Enable bit ((EVCTRL.TAMPEVEI) ) into the Event Control register (EVCTRL) enables the action on input events. Writing a '0' to this bit disables the action on input events.

### 23.6.6. 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.

### 23.6.7. 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


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

- 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'

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.

## 23.6.8. Additional Features

### 23.6.8.1. 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 (2 to 9 bits) 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:

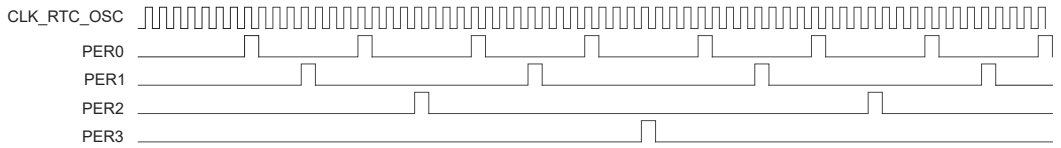
$$f_{PERIODIC(n)} = \frac{f_{CLK_RTC_OSC}}{2^{n+3}}$$

Where,

- $f_{CLK_RTC_OSC}$ is the frequency of the internal prescaler clock CLK_RTC_OSC
- 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, and so on. This is shown in the figure below.

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 23-5.** Example Periodic Events





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

### 23.6.8.2. 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 1 ppm 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 0.95367 ppm.

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.

### 23.6.8.3. 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.

When the COMPARE/ALARM register is written, the write buffer hold temporarily the COMPARE/ALARM value until the synchronization is complete (the SYNCBUSY.COMPn bit going to 0). After the write is completed the write buffer can be used as a odd general purpose register without 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 temporarily use the write buffer until the synchronization is complete (the SYNCBUSY.GPn bit going to 0). Therefore 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 ‘0’ 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.


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

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 23-2.** 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 | ALARM0 write buffer | GPn+1 |
| GPn+1 | COMPn/2 read buffer | (COMPn , COMPn+1) read buffer | ALARM0 read buffer | - |

### 23.6.8.4. Tamper Detection

The RTC provides eight 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 connection between IN and OUT can be made externally over the PCB, or internally over the TrustRAM. 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.

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).

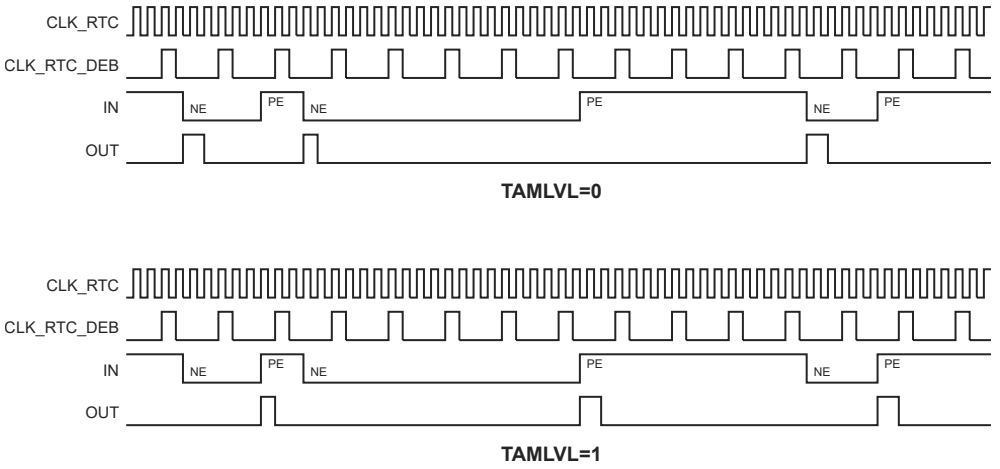
Up to eight external inputs (INn) can be used for tamper detection. The polarity for each input is selected with the Tamper Level bits in the Tamper Control register (TAMPCTRL.TAMPLVLn).

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 the following figure for an example.


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

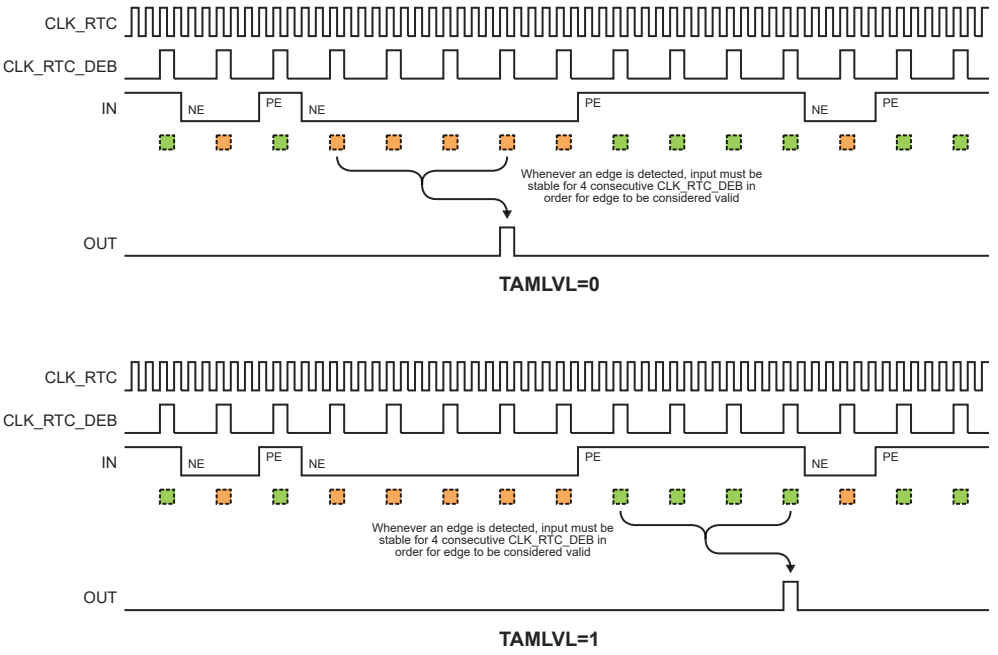
**Figure 23-6.** Edge Detection with Debouncer Disabled




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 the following figure for an example.

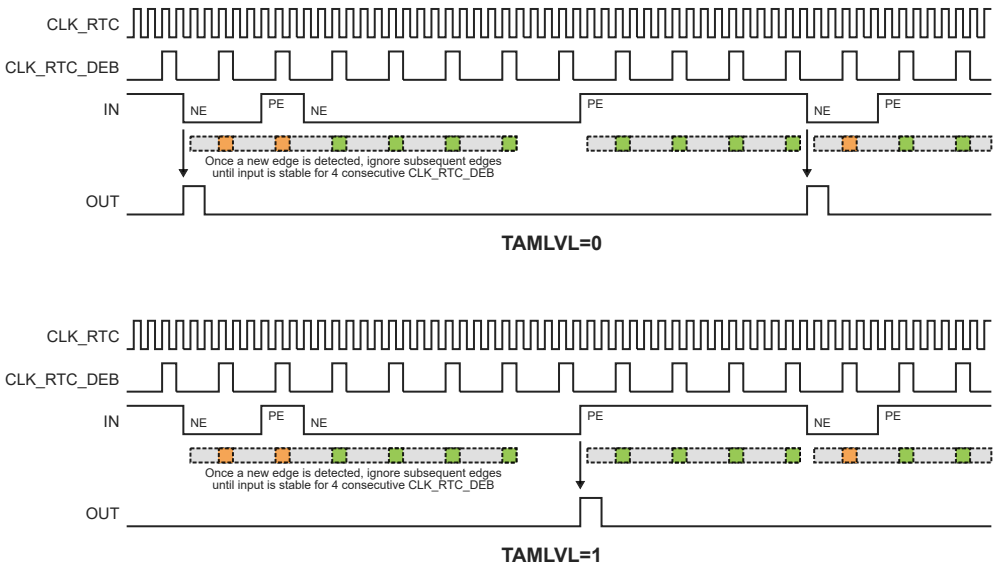
**Figure 23-7.** Edge Detection with Synchronous Stability Debouncing




- **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 the following figure for an example.


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

**Figure 23-8.** Edge Detection with Asynchronous Stability Debouncing




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 the following figure for an example.


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

**Figure 23-9.** Edge Detection with Majority Debouncing

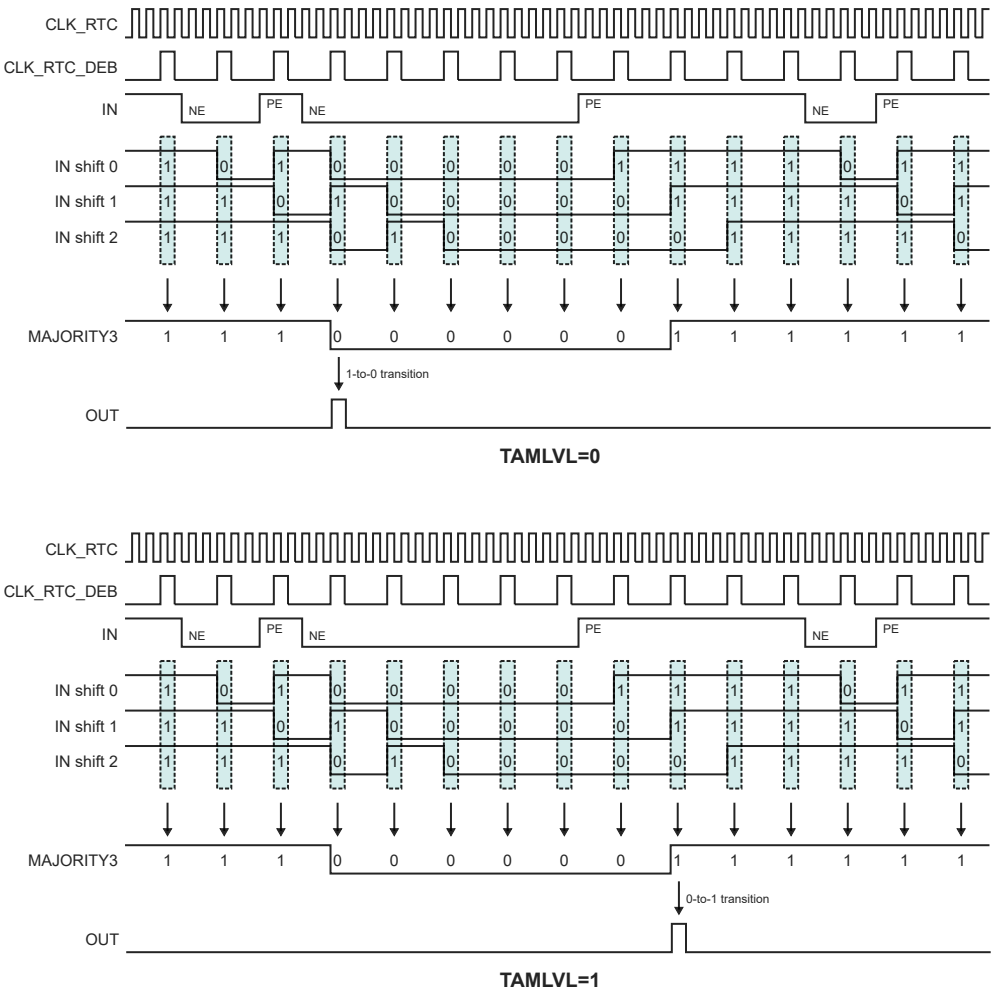



**Table 23-3.** Debouncer Configuration

| 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'). |

### 23.6.8.4.1. 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


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

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 `CAPTURE' or `ACTL'. 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.

#### 23.6.8.4.2. 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 RTC INn input pin. 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).

The selection of the TrustRAM active layer protection is done by setting the Active Layer Internal Select n bit in the TAMPCTRLB register (TAMPCTRLB.ALSIn), while clearing this bit selects the RTC INn and OUTn external protection.

**Note:** Only one tamper channel must be configured on the TrustRAM, while remaining channels can be independently configured on tamper detection pins.

**Note:** The Active Layer Protection works with distinct OUTn/INn pins when the Separate Tamper Output bit CTRLB.SEPTO is set. Alternatively, it can work with one output pin (OUT0) and multiple input pin INn. This is achieved by clearing the Separate Tamper Output bit CTRLB.SEPTO. If the TrustRAM active layer protection is selected for a channel and CTRLB.SEPTO is cleared, it will use the same internal signal as OUT0, otherwise it uses the OUTn signal.

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 rising edge of CLK_RTC and compared to the expected value of OUTn. Therefore up to one CLK_RTC period is available for propagation delay through the trace.
- Select Active Layer Monitoring Source (TrustRAM or INn/OUTn tamper pins) using the ALSIn bit of the TAMPCTRLB register.
- Enable Active Layer Protection by setting the CTRLB.RTCOUT bit.


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

## 23.7. Peripheral Dependencies

**Table 23-4.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| RTC | 0x45006000 | 12 : TAMPER<br>13 : OVF<br>14 : PER0/ PER1/PER2/ PER3/PER4/ PER5/PER6/ PER7<br>15 : CMP0/ CMP1/CMP2/ CMP3 | **APB:**<br>CLKMSK[0]<br>MASK12 | --- | 12<br>STATUS0<br>PERID12 | 3 : TIMESTAMP | Generators:<br>**PER0:**<br>CHANNEL[6]<br>**PER1:**<br>CHANNEL[7]<br>**PER2:**<br>CHANNEL[8]<br>**PER3:**<br>CHANNEL[9]<br>**PER4:**<br>CHANNEL[10]<br>**PER5:**<br>CHANNEL[11]<br>**PER6:**<br>CHANNEL[12]<br>**PER7:**<br>CHANNEL[13]<br>**CMP0:**<br>CHANNEL[14]<br>**CMP1:**<br>CHANNEL[15]<br>**CMP2:**<br>CHANNEL[16]<br>**CMP3:**<br>CHANNEL[17]<br>**TAMPER:**<br>CHANNEL[18]<br>**OVF:**<br>CHANNEL[19]<br>**PERD:**<br>CHANNEL[20]<br><br>Users:<br>**TAMPER:**<br>USER[1],Async | PD_CORE_BU |


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

## 23.8. Register Summary - MODE0

| 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 | SEPTO | ACTF[2:0] | | | | DEBF[2:0] | | |
| 0x04 | EVCTRL | 7:0 | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| | | 15:8 | OVFEO | TAMPEREO | | | | | CMPEO[1:0] | |
| | | 23:16 | | | | | | | | TAMPEVEI |
| | | 31:24 | | | | | | | | PERDEO |
| 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 ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | SYNCBUSY | 7:0 | | COMP1 | COMP0 | | COUNT | FREQCORR | ENABLE | SWRST |
| | | 15:8 | COUNTSYNC | | | | | | | |
| | | 23:16 | | | | | G3 | G2 | G1 | G0 |
| | | 31:24 | | | | | | | | |
| 0x14 ... 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 | COMP | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| | | 23:16 | COMP[23:16] | | | | | | | |
| | | 31:24 | COMP[31:24] | | | | | | | |
| 0x24 | COMP | 7:0 | COMP[7:0] | | | | | | | |
| | | 15:8 | COMP[15:8] | | | | | | | |
| | | 23:16 | COMP[23:16] | | | | | | | |
| | | 31:24 | COMP[31:24] | | | | | | | |
| 0x28 ... 0x63 | Reserved | | | | | | | | | |
| 0x64 | TIMESTAMP | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | COUNT[23:16] | | | | | | | |
| | | 31:24 | COUNT[31:24] | | | | | | | |


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

### 23.8.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x0000
**Property:** Write-Protected, Enable-Protected, Write-Synchronized

**Table 23-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | 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.

**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 |


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

| Value | Name | Description |
|---|---|---|
| 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 |

**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 |

**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 The 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 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.
This bit is not enable-protected.

**Notes:**

1. When the CTRLA.SWRST is written, the user must poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.

| Value | Description |
|---|---|
| 0 | has no effect |
| 1 | The reset operation is ongoing |


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

### 23.8.2. Control B

**Name:** CTRLB
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

**Table 23-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SEPTO | ACTF[2:0] | | | | DEBF[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 15 – SEPTO** Separate Tamper Outputs

**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.


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

| 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. |
| 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.

### 23.8.3. Event Control

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** RW

**Table 23-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PERDEO |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| 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 | | | | | CMPEO[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 24 – PERDEO** Periodic Interval Daily Event Output Enable

| Value | Description |
|---|---|
| 0 | Periodic Daily event is disabled and will not be generated. |
| 1 | Periodic Daily event is enabled and will be generated.<br>The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000). |

**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


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

| 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 9:8 – CMPEO[1:0]** Compare x Event Output Enable

| 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. |
| 1 | Periodic Interval n event is enabled and will be generated. |


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

### 23.8.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x08
**Reset:** 0x0000
**Property:** RW

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.

**Table 23-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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.


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

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.

### 23.8.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x0A
**Reset:** 0x0000
**Property:** RW

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.

**Table 23-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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.


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

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.

### 23.8.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** RW

**Note:** Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 23-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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
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 – 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.

### 23.8.7. Synchronization Busy Status

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** R

**Table 23-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | G3 | G2 | G1 | G0 |
| 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 – GP** General Purpose x Register Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for GPn register is complete. |
| 1 | Write synchronization for GPn register is ongoing. |

**Bit 15 – COUNTSYNC** Count Synchronization Enable Bit Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for the CTRLA.COUNTSYNC bit is complete. |
| 1 | Write synchronization for the 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 Register Busy

| Value | Description |
|---|---|
| 0 | Read/write synchronization for the COUNT register is complete. |


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

| Value | Description |
|---|---|
| 1 | Read/write synchronization for the COUNT register is ongoing. |

**Bit 2 – FREQCORR** FREQCORR Register Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for the FREQCORR register is complete. |
| 1 | Write synchronization for the FREQCORR register is ongoing. |

**Bit 1 – ENABLE** Enable Bit Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for the CTRLA.ENABLE bit is complete. |
| 1 | Write synchronization for the CTRLA.ENABLE bit is ongoing. |

**Bit 0 – SWRST** Software Reset Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for the CTRLA.SWRST bit is complete. |
| 1 | Write synchronization for the CTRLA.SWRST bit is ongoing. |


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

### 23.8.8. Counter Value

**Name:** COUNT
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

**Table 23-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

**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.

### 23.8.9. Compare n Value in COUNT32 mode (CTRLA.MODE=0)

**Name:** COMP
**Offset:** 0x20 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** RW

**Table 23-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 COMP0 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 '1'.


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

### 23.8.10. Timestamp

**Name:** TIMESTAMP
**Offset:** 0x64
**Reset:** 0x00000000
**Property:** -

**Table 23-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[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 |
|---|---|---|---|---|---|---|---|---|
| | COUNT[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 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.

## 23.9. Register Summary - MODE1

| 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 | SEPTO | ACTF[2:0] | | | | DEBF[2:0] | | |
| 0x04 | EVCTRL | 7:0 | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| | | 15:8 | OVFEO | TAMPEREO | | | CMPEO0[3:0] | | | |
| | | 23:16 | | | | | | | | TAMPEVEI |
| | | 31:24 | | | | | | | | PERDEO |
| 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 ... 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 ... 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 ... 0x63 | Reserved | | | | | | | | | |
| 0x64 | TIMESTAMP | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 23.9.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x0000
**Property:** Write-Protected, Enable-Protected, Write-Synchronized

**Table 23-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | 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

**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 |
|---|---|---|
| 0 | OFF | CLK_RTC_CNT = GCLK_RTC/1 |
| 0xA | DIV512 | CLK_RTC_CNT = GCLK_RTC/512 |
| 0xB | DIV1024 | CLK_RTC_CNT = GCLK_RTC/1024 |
| 1 | DIV1 | CLK_RTC_CNT = GCLK_RTC/1 |
| 2 | DIV2 | CLK_RTC_CNT = GCLK_RTC/2 |
| 3 | DIV4 | CLK_RTC_CNT = GCLK_RTC/4 |


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

| Value | Name | Description |
|---|---|---|
| 4 | DIV8 | CLK_RTC_CNT = GCLK_RTC/8 |
| 5 | DIV16 | CLK_RTC_CNT = GCLK_RTC/16 |
| 6 | DIV32 | CLK_RTC_CNT = GCLK_RTC/32 |
| 7 | DIV64 | CLK_RTC_CNT = GCLK_RTC/64 |
| 8 | DIV128 | CLK_RTC_CNT = GCLK_RTC/128 |
| 9 | DIV256 | CLK_RTC_CNT = GCLK_RTC/256 |

**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 |
|---|---|---|
| 0 | COUNT32 | Mode 0: 32-bit Counter |
| 1 | COUNT16 | Mode 1: 16-bit Counter |
| 2 | CLOCK | Mode 2: Clock/Calendar |

**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 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.

This bit is not enable-protected.

| 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.

### 23.9.2. Control B

**Name:** CTRLB
**Offset:** 0x02
**Reset:** 0x0000
**Property:** RW

**Table 23-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SEPTO | ACTF[2:0] | | | | DEBF[2:0] | | |
| Access | RW | 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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 15 – SEPTO** Separate Tamper Outputs

**Bits 14:12 – ACTF[2:0]** Active Layer Frequency

| 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

| 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

| 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. |


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

**Bit 6 – RTCOUT** RTC Output Enable

| Value | Description |
|---|---|
| 0 | The RTC active layer output is disabled. |
| 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.

### 23.9.3. Event Control

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** RW

**Table 23-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PERDEO |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| 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 | | | CMPEOn[3:0] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | 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 24 – PERDEO** Periodic Interval Daily Event Output Enable

| Value | Description |
|---|---|
| 0 | Periodic Daily event is disabled and will not be generated. |
| 1 | Periodic Daily event is enabled and will be generated.<br>The event occurs at the overflow of the RTC counter (i.e., when the RTC counter goes from 0xFFFF to 0x0000). |

**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


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

| 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 11:8 – CMPEOn[3:0]** 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. |
| 1 | Periodic Interval n event is enabled and will be generated. |


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

### 23.9.4. Interrupt Enable Clear

**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.

**Table 23-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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. |


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

| Value | Description |
|---|---|
| 1 | Periodic Interval n interrupt is enabled. |


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

### 23.9.5. Interrupt Enable Set

**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.

**Table 23-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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. |


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

| Value | Description |
|---|---|
| 1 | Periodic Interval n interrupt is enabled. |


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

### 23.9.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** -

**Table 23-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**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.

| Value | Description |
|---|---|
| 0 | The Tamper interrupt it disabled. |
| 1 | The Tamper interrupt is enabled. |

**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.

| Value | Description |
|---|---|
| 0 | The Compare n interrupt is disabled. |
| 1 | The Compare n interrupt is enabled. |


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

**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.

| 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.

### 23.9.7. Synchronization Busy Status

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** R

**Table 23-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Synchronization Enable Bit Busy

| 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** PER Register Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for PER register is complete. |


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

| Value | Description |
|---|---|
| 1 | Write synchronization for PER register is ongoing. |

**Bit 3 – COUNT** COUNT Register Busy

| Value | Description |
|---|---|
| 0 | Read/write synchronization for COUNT register is complete. |
| 1 | Read/write synchronization for COUNT register is ongoing. |

**Bit 2 – FREQCORR** FREQCORR Register Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for FREQCORR register is complete. |
| 1 | Write synchronization for FREQCORR register is ongoing. |

**Bit 1 – ENABLE** Enable Bit Busy

| 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 Bit Busy

| 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.

### 23.9.8. Counter Value in COUNT16 mode (CTRLA.MODE=1)

**Name:** COUNT
**Offset:** 0x18
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

**Table 23-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

**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.

### 23.9.9. Counter Period in COUNT16 mode (CTRLA.MODE=1)

**Name:** PER
**Offset:** 0x1C
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

**Table 23-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

**Note:** This register is write-synchronized: SYNCBUSY.PER must be checked to ensure the PER register synchronization is complete.

| 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.

### 23.9.10. Compare n Value in in COUNT16 mode (CTRLA.MODE=1)

**Name:** COMP
**Offset:** 0x20 + n*0x02 [n=0..3]
**Reset:** 0x0000
**Property:** RW

**Table 23-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 23.9.11. Timestamp

**Name:** TIMESTAMP
**Offset:** 0x64
**Reset:** 0x00000000
**Property:** -

**Table 23-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

## 23.10. Register Summary - MODE2

| 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 | SEPTO | ACTF[2:0] | | | | DEBF[2:0] | | |
| 0x04 | EVCTRL | 7:0 | PEREO7 | PEREO6 | PEREO5 | PEREO4 | PEREO3 | PEREO2 | PEREO1 | PEREO0 |
| | | 15:8 | OVFEO | TAMPEREO | | | | | ALARMEO0[1:0] | |
| | | 23:16 | | | | | | | | TAMPEVEI |
| | | 31:24 | | | | | | | | PERDEO |
| 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 ... 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 ... 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 ... 0x63 | Reserved | | | | | | | | | |
| 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] | |


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

### 23.10.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x0000
**Property:** Write-Protected, Enable-Protected, Write-Synchronized

**Table 23-26.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | 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 no synchronized.

**Bit 13 – BKTRST** BKUP Registers Reset On Tamper Enable

**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 |


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

| Value | Name | Description |
|---|---|---|
| 0x9 | DIV256 | CLK_RTC_CNT = GCLK_RTC/256 |
| 0xA | DIV512 | CLK_RTC_CNT = GCLK_RTC/512 |
| 0xB | DIV1024 | CLK_RTC_CNT = GCLK_RTC/1024 |

**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 |
|---|---|---|
| 0 | COUNT32 | Mode 0: 32-bit Counter |
| 1 | COUNT16 | Mode 1: 16-bit Counter |
| 2 | CLOCK | Mode 2: Clock/Calendar |

**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 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.
This bit is not enable-protected.

| 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.

### 23.10.2. Control B

**Name:** CTRLB
**Offset:** 0x02
**Reset:** 0x0000
**Property:** RW

**Table 23-27.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | SEPTO | ACTF[2:0] | | | | DEBF[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 15 – SEPTO** Separate Tamper Outputs

| Value | Description |
|---|---|
| 0 | IN[n] is compared tp OUT[0] (backward-compatible). |
| 1 | IN[n] is compared tp OUT[n]. |

**Bits 14:12 – ACTF[2:0]** Active Layer Frequency

| 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

| 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


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

| 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. |
| 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.

### 23.10.3. Event Control in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** EVCTRL
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 23-28.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PERDEO |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| 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 | | | | | ALARMEOn[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 24 – PERDEO** Periodic Interval Daily Event Output Enable

| Value | Description |
|---|---|
| 0 | Periodic Daily event is disabled and will not be generated. |
| 1 | Periodic Daily event is enabled and will be generated.<br>The event occurs at the last second of each day depending on the CTRLA.CLKREP bit:<br>• If CLKREP = 0, the event will occur at 23:59:59<br>• If CLKREP = 1, the event will occur at 11:59:59, PM = 1 |

**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. |


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

**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 9:8 – ALARMEOn[1:0]** 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. |
| 1 | Periodic Interval n event is enabled and will be generated. |


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

### 23.10.4. Interrupt Enable Clear

**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.

**Table 23-29.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Enable

| 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 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.

### 23.10.5. Interrupt Enable Set

**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.

**Table 23-30.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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. |


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

| Value | Description |
|---|---|
| 1 | Periodic Interval n interrupt is enabled. |


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

### 23.10.6. Interrupt Flag Status and Clear in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** INTFLAG
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** -

**Table 23-31.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | OVF | TAMPER | | | | | ALARM1 | ALARM0 |
| Access | RW | 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.

| Value | Description |
|---|---|
| 0 | The Overflow interrupt is disabled. |
| 1 | The Overflow interrupt is enabled. |

**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.

| Value | Description |
|---|---|
| 0 | The Tamper interrupt it disabled. |
| 1 | The Tamper interrupt is enabled. |

**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.

| Value | Description |
|---|---|
| 0 | The Alarm n interrupt is disabled. |
| 1 | The Alarm n interrupt is enabled. |


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

**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.

| 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.

### 23.10.7. Synchronization Busy Status

**Name:** SYNCBUSY
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** -

**Table 23-32.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Synchronization Enable Bit Busy

| 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. |


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

| Value | Description |
|---|---|
| 1 | Write synchronization for ALARMx register is ongoing. |

**Bit 3 – CLOCK** CLOCK Register Busy

| Value | Description |
|---|---|
| 0 | Read/write synchronization for CLOCK register is complete. |
| 1 | Read/write synchronization for CLOCK register is ongoing. |

**Bit 2 – FREQCORR** FREQCORR Register Busy

| Value | Description |
|---|---|
| 0 | Write synchronization for FREQCORR register is complete. |
| 1 | Write synchronization for FREQCORR register is ongoing. |

**Bit 1 – ENABLE** Enable Bit Busy

| 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 Bit Busy

| 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.

### 23.10.8. Clock Value in Clock/Calendar mode (CTRLA.MODE=2)

**Name:** CLOCK
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized

**Table 23-33.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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


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

**Bits 5:0 – SECOND[5:0]** Second
0 – 59


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

### 23.10.9. Timestamp

**Name:** TIMESTAMP
**Offset:** 0x64
**Reset:** 0x00000000
**Property:** R

**Table 23-34.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Timestamp Value
The year value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 25:22 – MONTH[3:0]** Month Timestamp Value
The month value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 21:17 – DAY[4:0]** Day Timestamp Value
The day value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 16:12 – HOUR[4:0]** Hour Timestamp Value
The hour value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 11:6 – MINUTE[5:0]** Minute Timestamp Value
The minute value is captured by the TIMESTAMP when a tamper condition occurs.

**Bits 5:0 – SECOND[5:0]** Second Timestamp Value
The second value is captured by the TIMESTAMP when a tamper condition occurs.


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

## 23.11. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0x0D | Reserved | | | | | | | | | |
| 0x0E | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0F ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | FREQCORR | 7:0 | SIGN | VALUE[6:0] | | | | | | |
| 0x15 ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | ALARM - MODE2_ALARM Mode | 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 | MASK - MODE2_ALARM Mode | 7:0 | | | | | | SEL[2:0] | | |
| 0x25 ... 0x3F | Reserved | | | | | | | | | |
| 0x40 | G0 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x44 | G1 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x48 | G2 | 7:0 | GP[7:0] | | | | | | | |
| | | 15:8 | GP[15:8] | | | | | | | |
| | | 23:16 | GP[23:16] | | | | | | | |
| | | 31:24 | GP[31:24] | | | | | | | |
| 0x4C | G3 | 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 | IN7ACT[1:0] | | IN6ACT[1:0] | | IN5ACT[1:0] | | IN4ACT[1:0] | |
| | | 23:16 | | | | TAMLVL4 | TAMLVL3 | TAMLVL2 | TAMLVL1 | TAMLVL0 |
| | | 31:24 | | | | DEBNC4 | DEBNC3 | DEBNC2 | DEBNC1 | DEBNC0 |
| 0x64 ... 0x67 | Reserved | | | | | | | | | |
| 0x68 | TAMPID | 7:0 | | | | TAMPID4 | TAMPID3 | TAMPID2 | TAMPID1 | TAMPID0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TAMPEVT | | | | | | | |
| 0x6C | TAMPCTRLB | 7:0 | ALSI[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 23.11.1. Debug Control

**Name:** DBGCTRL
**Offset:** 0x0E
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 23-35.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 0 – DBGRUN** Run During Debug
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.

### 23.11.2. Frequency Correction

**Name:** FREQCORR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

**Table 23-36.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

**Note:** This register is write-synchronized: SYNCBUSY.FREQCORR must be checked to ensure the FREQCORR register synchronization is complete.

| 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

| 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.

### 23.11.3. Alarm n Value

**Name:** ALARM - MODE2_ALARM Mode
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** RW

**Table 23-37.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | YEAR[5:0] | | | | | | MONTH[3:2] | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | MONTH[1:0] | | DAY[4:0] | | | | | HOUR[4] |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | HOUR[3:0] | | | | MINUTE[5:2] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | MINUTE[1:0] | | SECOND[5:0] | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:26 – YEAR[5:0]** Year

**Bits 25:22 – MONTH[3:0]** Month

**Bits 21:17 – DAY[4:0]** Day

**Bits 16:12 – HOUR[4:0]** Hour

| Value | Name | Description |
|---|---|---|
| 0x00 | AM | Morning hour |
| 0x10 | PM | Afternoon hour |

**Bits 11:6 – MINUTE[5:0]** Minute

**Bits 5:0 – SECOND[5:0]** Second


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

### 23.11.4. Alarm n Mask

**Name:** MASK - MODE2_ALARM Mode
**Offset:** 0x24
**Reset:** 0x00
**Property:** RW

**Table 23-38.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bits 2:0 – SEL[2:0]** Alarm Mask Selection

| 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 |


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

### 23.11.5. General Purpose

**Name:** GP
**Offset:** 0x40 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** RW

**Table 23-39.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 23.11.6. Tamper Control

**Name:** TAMPCTRL
**Offset:** 0x60
**Reset:** 0x00000000
**Property:** RW

**Table 23-40.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | IN7ACT[1:0] | | IN6ACT[1:0] | | IN5ACT[1:0] | | IN4ACT[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 |
|---|---|---|---|---|---|---|---|---|
| | IN3ACT[1:0] | | IN2ACT[1:0] | | IN1ACT[1:0] | | IN0ACT[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 24, 25, 26, 27, 28 – DEBNCn** Debounce Enable of Tamper Input INn [n=0..3]
**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 – TAMLVLn** Tamper Level Select of Tamper Input INn [n=0..3]
**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 15:14 – IN7ACT[1:0]** Tamper Input 7 Action

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | Off (Disabled) |
| 0x1 | WAKE | Wake and set Tamper flag |
| 0x2 | CAPTURE | Capture timestamp and set Tamper flag |


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

| Value | Name | Description |
|---|---|---|
| 0x3 | ACTL | Compare IN7 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |

**Bits 13:12 – IN6ACT[1:0]** Tamper Input 6 Action

| 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 IN6 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |

**Bits 11:10 – IN5ACT[1:0]** Tamper Input 5 Action

| 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 IN5 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |

**Bits 9:8 – IN4ACT[1:0]** Tamper Input 4 Action

| 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 IN4 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |

**Bits 7:6 – IN3ACT[1:0]** Tamper Input 3 Action

| 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 IN3 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |

**Bits 5:4 – IN2ACT[1:0]** Tamper Input 2 Action

| 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 IN2 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |

**Bits 3:2 – IN1ACT[1:0]** Tamper Input 1 Action

| 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 IN1 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |

**Bits 1:0 – IN0ACT[1:0]** Tamper Input 0 Action

| 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 IN0 to OUT. When a mismatch occurs, capture timestamp and set Tamper flag |


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

### 23.11.7. Tamper ID

**Name:** TAMPID
**Offset:** 0x68
**Reset:** 0x00000000
**Property:** RW

**Table 23-41.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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

| 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 – TAMPIDn** Tamper Input x 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.

### 23.11.8. Tamper Control B

**Name:** TAMPCTRLB
**Offset:** 0x6C
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 23-42.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | ALSI[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 7:0 – ALSI[7:0]** Active Layer Select Internal x

| Value | Description |
|---|---|
| 0 | Active layer Protection is monitoring the RTC signal using INn and OUTn tamper pins |
| 1 | Active layer Protection is monitoring the RTC signal on the TrustRAM shield |


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

# 24. Direct Memory Access Controller (DMAC)

## 24.1. 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 all can receive different types of transfer triggers to generate transfer requests from the DMA channels to the arbiter, see also the Block Diagram. The arbiter will grant 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 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 the following four bus interfaces:

- Data transfer bus – Used for performing the actual DMA transfer.
- AHB/APB Bridge bus – Used when writing and reading the I/O registers of the DMAC.
- Descriptor fetch bus – Used by the fetch engine to fetch transfer descriptors before data transfer can be started or continued.
- Write-back bus – Used to write the transfer descriptor back to SRAM.

All buses are AHB host interfaces except the AHB/APB Bridge bus, which is an APB client interface.

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.

## 24.2. Features

The following are key features of the DMAC module:

- 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
  - Multi-buffer or circular buffer modes by linking multiple descriptors
- Up to 8 channels

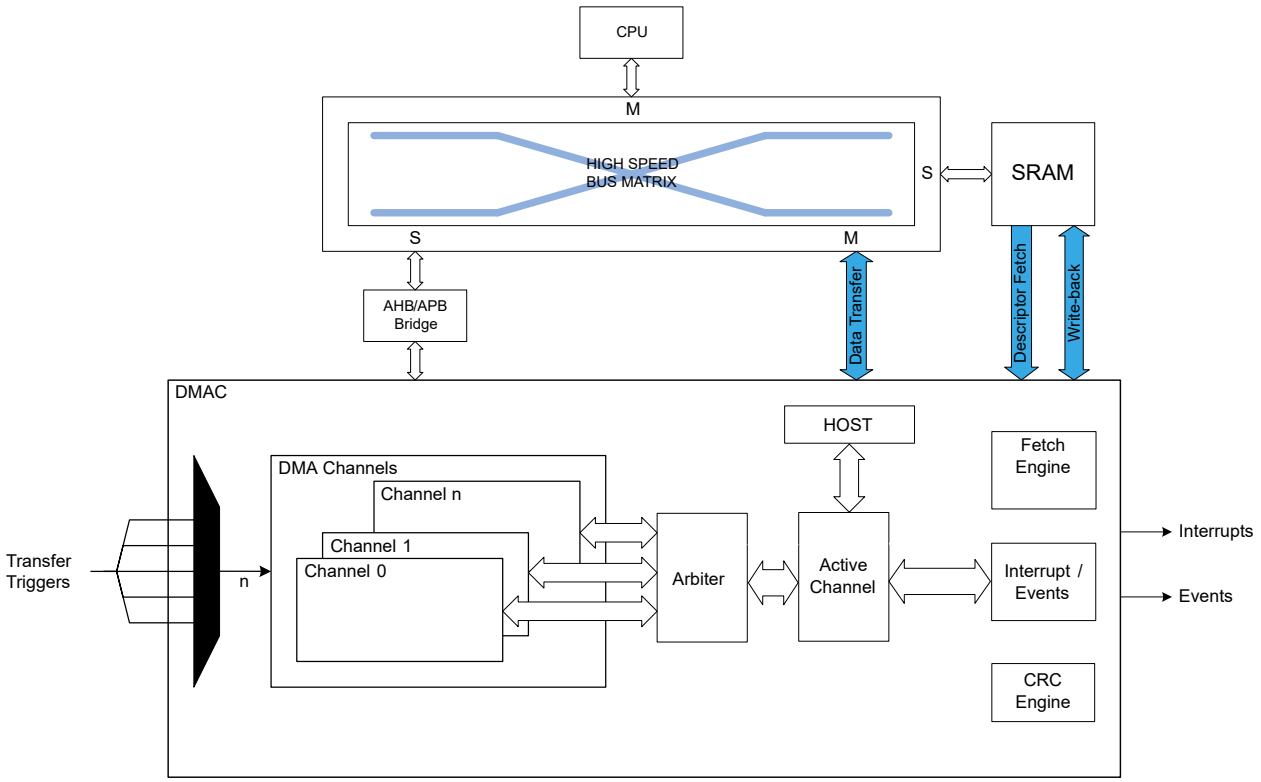

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

  - Enable 8 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 KB to 256 KB 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
- 4 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 8 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
  - CRC-16 (CRC-CCITT)
  - CRC-32 (IEEE® 802.3)


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

## 24.3. Block Diagram

**Figure 24-1.** DMAC Block Diagram




## 24.4. Product Dependencies

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

### 24.4.1. 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.

### 24.4.2. 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.

### 24.4.3. Interrupt

The interrupt request line is connected to the interrupt controller. Using the DMAC interrupt requires the interrupt controller to be configured first.

### 24.4.4. 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.


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

### 24.4.5. 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.

## 24.5. Functional Description

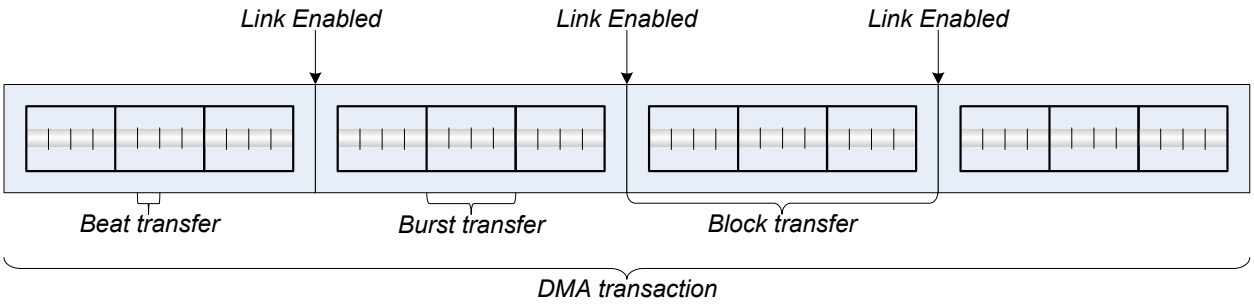
### 24.5.1. Principle of Operation

The DMAC consists of a DMA module and a CRC module.

#### 24.5.1.1. 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 24-2.** 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)
- 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. .
- 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.


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

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 of the configuration, the DMA channel will either be suspended or disabled.

#### 24.5.1.2. 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 section for details.

### 24.5.2. Basic Operation

#### 24.5.2.1. Initialization

The following DMAC registers are enable-protected, meaning that they can only be written when the DMAC is disabled (CTRL.DMAENABLE = 0):

- Descriptor Base Memory Address register (BASEADDR)
- Write-Back Memory Base Address register (WRBADDR)

The following DMAC bit is enable-protected, meaning that it can only be written when both the DMAC and CRC are disabled (CTRL.DMAENABLE = 0 and CTRL.CRCENABLE = 0):

- Software Reset bit in Control register (CTRL.SWRST)

The following DMA channel register is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled (CHCTRLA.ENABLE = 0):

- Channel Control B (CHCTRLB) register, except the Command bit (CHCTRLB.CMD) and the Channel Arbitration Level bit (CHCTRLB.LVL)

The following DMA channel bit is enable-protected, meaning that it can only be written when the corresponding DMA channel is disabled:

- Channel Software Reset bit in Channel Control A register (CHCTRLA.SWRST)

The following CRC registers are enable-protected, meaning that they can only be written when the CRC is disabled (CTRL.CRCENABLE = 0):

- CRC Control register (CRCCTRL)
- CRC Checksum register (CRCCHKSUM)

Enable-protection is denoted by the "Enable-Protected" property in the register description.

Before the DMAC is enabled it must be configured, as outlined by the following steps:

- 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)

Before a DMA channel is enabled, the DMA channel and the corresponding first transfer descriptor must be configured, as outlined by the following steps:

- DMA channel configurations
  - The channel number of the DMA channel to configure must be written to the Channel ID (CHID) register
  - Trigger action must be selected by writing the Trigger Action bit group in the Channel Control B register (CHCTRLB.TRIGACT)


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

  - Trigger source must be selected by writing the Trigger Source bit group in the Channel Control B register (CHCTRLB.TRIGSRC)

- 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 register (BTCTRL.BEATSIZE)
  - The transfer descriptor must be made valid by writing a one to the Valid bit in the Block Transfer Control register (BTCTRL.VALID)
  - 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

If CRC calculation is needed, the CRC engine must be configured before it is enabled, as outlined by the following steps:

- The CRC input source must selected by writing the CRC Input Source bit group in the CRC Control register (CRCCTRL.CRCSRC)
- The type of CRC calculation must be selected by writing the CRC Polynomial Type bit group in the CRC Control register (CRCCTRL.CRCPOLY)
- 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 register (CRCCTRL.CRCBEATSIZE)

#### 24.5.2.2. Enabling, Disabling, and Resetting

The DMAC is enabled by writing the DMA Enable bit in the Control register (CTRL.DMAENABLE) to '1'. The DMAC is disabled by writing a '0' to CTRL.DMAENABLE.

A DMA channel is enabled by writing the Enable bit in the Channel Control A register (CHCTRLA.ENABLE) to '1', after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). A DMA channel is disabled by writing a '0' to CHCTRLA.ENABLE.

The CRC is enabled by writing a '1' to the CRC Enable bit in the Control register (CTRL.CRCENABLE). The CRC is disabled by writing a '0' to CTRL.CRCENABLE.

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 (CHCTRLA.SWRST), after writing the corresponding channel id to the Channel ID bit group in the Channel ID register (CHID.ID). 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.

#### 24.5.2.3. 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.

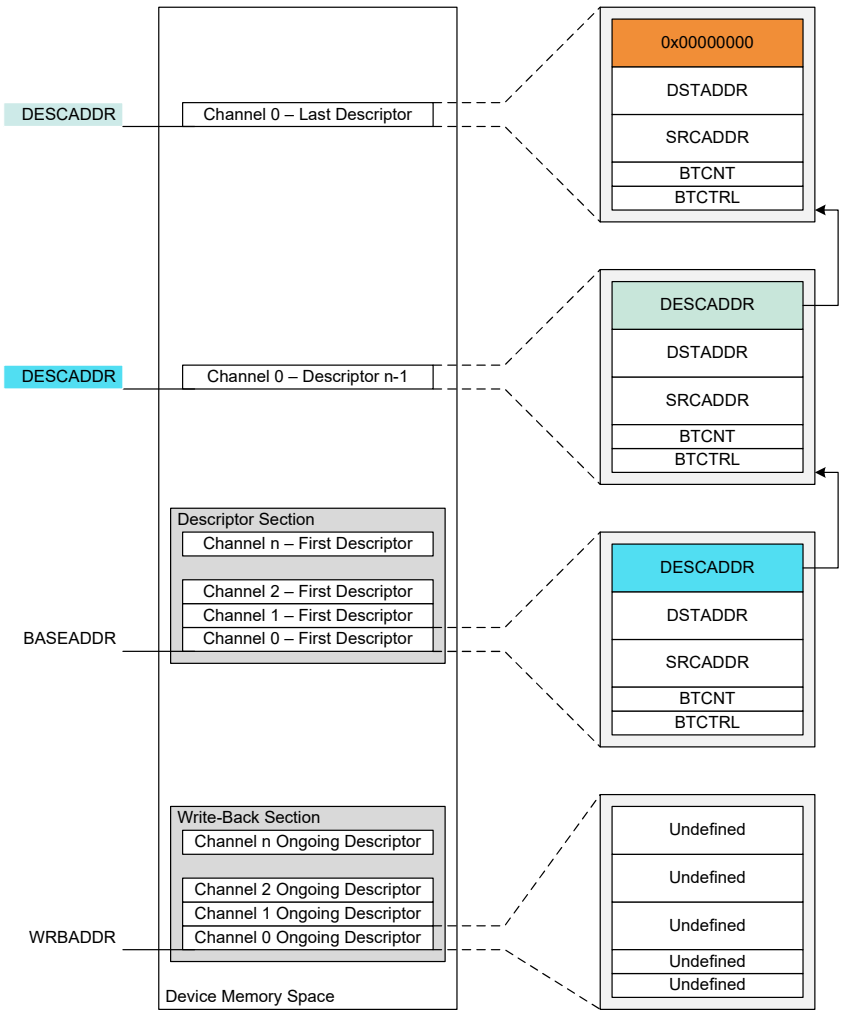
The descriptor memory section is where the DMAC expects to find the first transfer descriptors for all DMA channels. As BASEADDR points only to the first transfer descriptor of channel ‘0’ (see


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

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 the Linked descriptor.

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 descriptor.

**Figure 24-3.** 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


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

to modify the first transfer descriptor. The benefit of having descriptor memory and write-back memory in the same section is that it requires less SRAM.

### 24.5.2.4. 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 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 active channel is the DMA channel being granted access to perform its next transfer. When the arbiter has granted a DMA channel access to the DMAC, the corresponding bit PENDCH.PENDCHx will be cleared. See also the following figure.

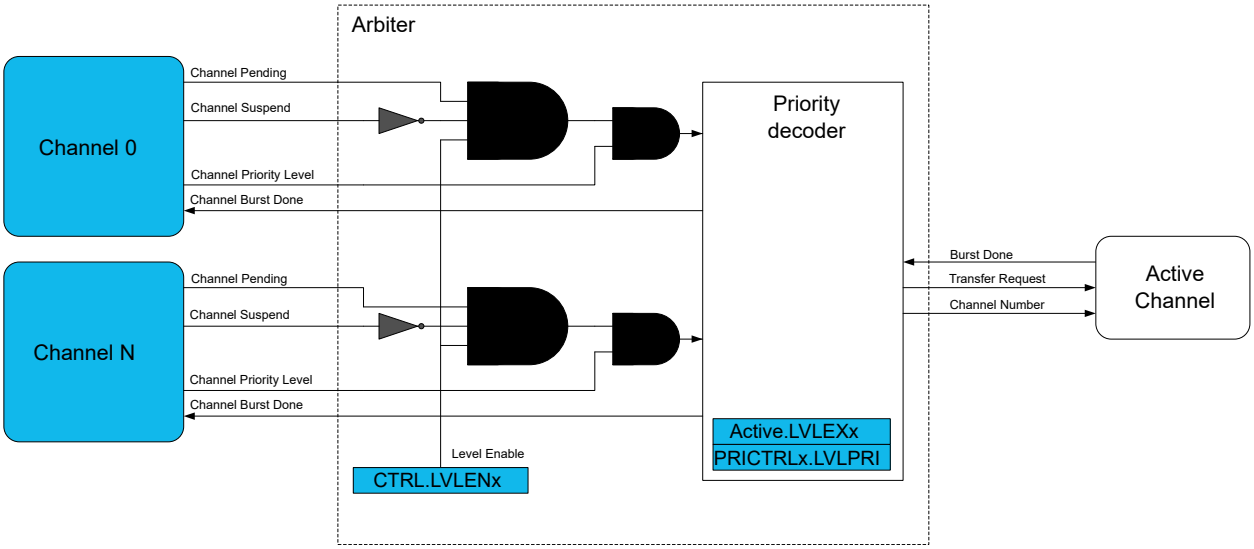
If the upcoming transfer is the first for the transfer request, the corresponding Busy Channel x bit in the Busy Channels register will be set (BUSYCHBUSYCHx=1), and it will remain '1' for the subsequent granted transfers.

When the channel has performed its granted 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 fed into the queue of channels with pending transfers, the corresponding BUSYCH.BUSYCHx will remain '1'. 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. 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 24-4.** 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 a 4-level priority scheme. The priority level for a channel is configured by writing to the Channel Arbitration Level bit group in the Channel Control B register


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

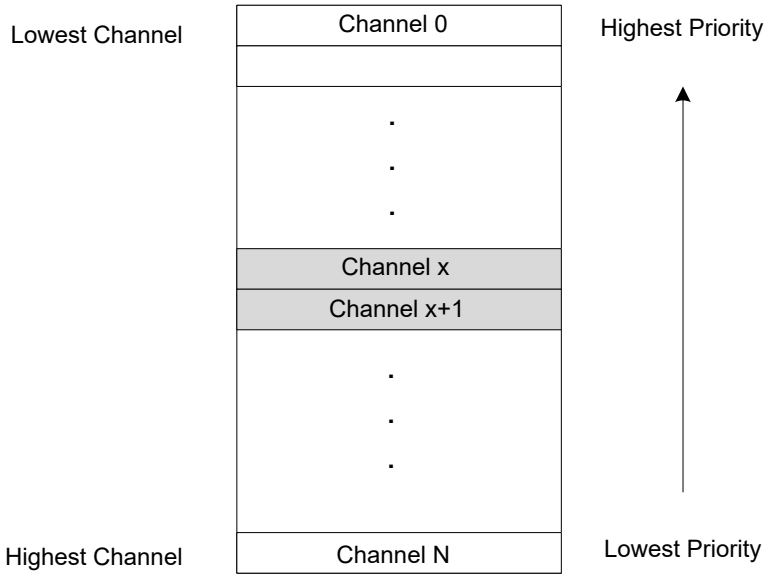
(CHCTRLB.LVL). As long as all priority levels are enabled, a channel with a higher priority level number will have priority over a channel with a lower priority level number. Each priority level x is enabled by setting the corresponding Priority Level x Enable bit in the Control register (CTRL.LVLENx=1).

Within each priority level the DMAC's arbiter can be configured to prioritize statically or dynamically:

*Static Arbitration* within a priority level is selected by writing a '0' to the Level x Round-Robin Scheduling Enable bit in the Priority Control 0 register (PRICTRL0.RRLVLENx).

When static arbitration is selected, the arbiter will prioritize a low channel number over a high channel number as shown in the figure below. When using the static arbitration 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 24-5.** Static Priority Scheduling

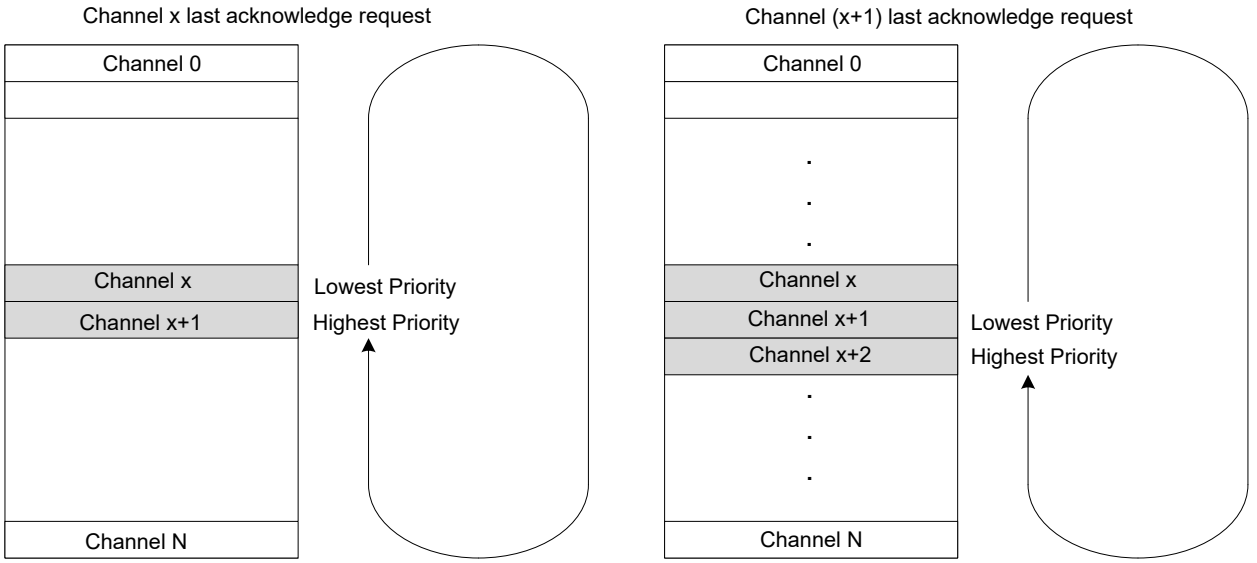



*Dynamic Arbitration* within a priority level is selected by writing a '1' to PRICTRL0.RRLVLENx.

The dynamic arbitration scheme in the DMAC is round-robin. 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 Priority Scheduling follwoung figure. 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 24-6.** Dynamic (Round-Robin) Priority Scheduling




### 24.5.2.5. 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 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 transfer, the optional output event Beat will be generated if configured and enabled, and the active channel will perform a new 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.


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

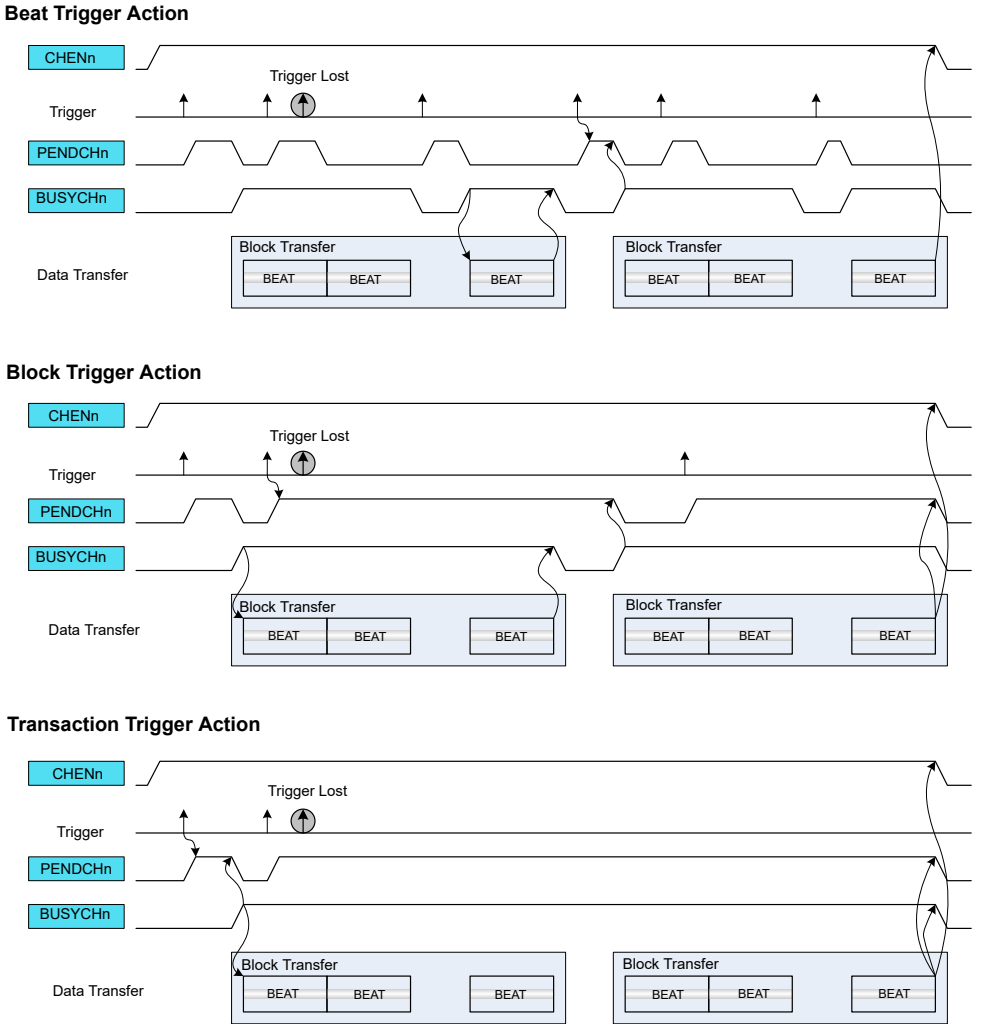
### 24.5.2.6. Transfer Triggers and Actions

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 Control B (CHCTRLB.TRIGSRC).

The trigger actions are available in the Trigger Action bit group in the Channel Control B register (CHCTRLB.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. If 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 beat transfer (CHCTRLB.TRIGACT=0x2) or transaction transfer (CHCTRLB.TRIGACT=0x3) instead of a block transfer (CHCTRLB.TRIGACT=0x0).

The followin image shows an example where triggers are used with two linked block descriptors.

**Figure 24-7.** 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 (CHSTATUS.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 Status register (CHSTATUS.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.

#### 24.5.2.7. Addressing

Each block transfer needs to have both 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:

$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right) \cdot 2^{\mathrm{STEPSIZE}}$$

IfBTCTRL.STEPSEL=0:

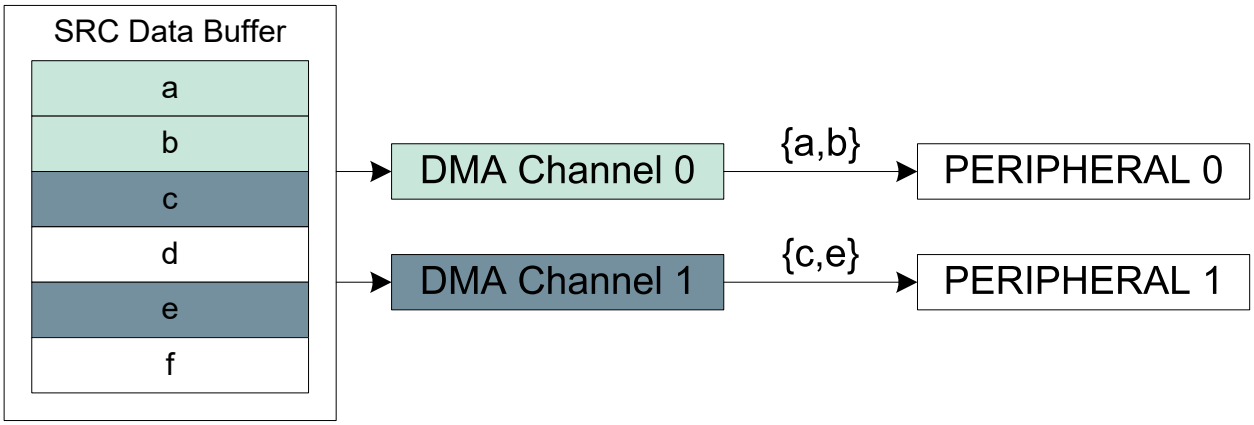
$$\mathrm{SRCADDR} = \mathrm{SRCADDR}_{START} + BTCNT \cdot \left(BEATSIZE + 1\right)$$

- $\mathrm{SRCADDR}_{\mathrm{START}}$ 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
- 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 24-8.** 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.

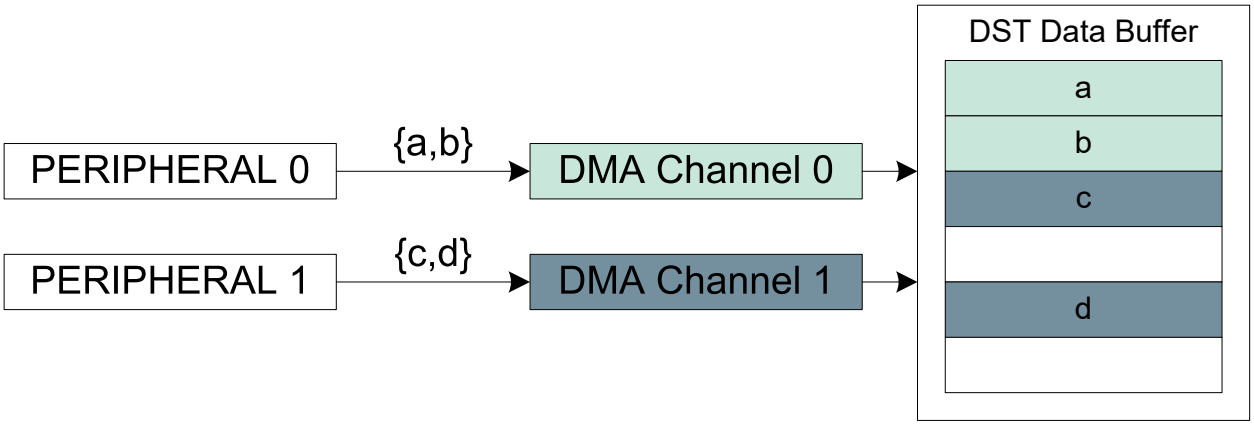
When the destination address incrementation is configured (BTCTRL.DSTINC=1), DSTADDR must be set and calculated as follows:

| | |
|---|---|
| $DSTADDR = DSTADDR_{START} + BTCNT \bullet \left(BEATSIZE + 1\right) \bullet 2^{STEPSIZE}$ | where BTCTRL.STEPSEL is zero |
| $DSTADDR = DSTADDR_{START} + BTCNT \bullet (BEATSIZE + 1)$ | where BTCTRL.STEPSEL is one |

- $DSTADDR_{START}$ 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

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).

**Figure 24-9.** Destination Address Increment





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

#### 24.5.2.8. Error Handling

If a bus error is received from an AHB slave 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.

### 24.5.3. Additional Features

#### 24.5.3.1. 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.

Memory sections 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 from the Next Descriptor Address (DESCADDR) register 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 additional information on how the next descriptor is fetched from SRAM, refer to the section *Data Transmission*.

##### 24.5.3.1.1. 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.

##### 24.5.3.1.2. 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


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

- 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.

#### 24.5.3.1.3. 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 these 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:
   a. Apply the software suspend command to the channel.
   b. Perform Step 2.1 through Step 2.4.
   c. Apply the software resume command to the channel.

### 24.5.3.2. 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.

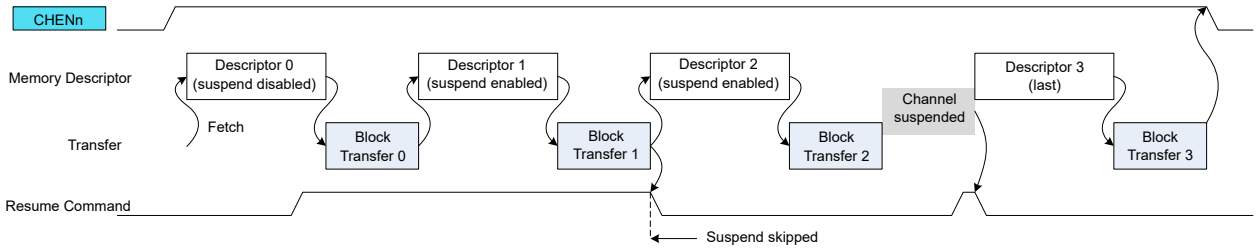
### 24.5.3.3. Channel Resume and Next Suspend Skip

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


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

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.

**Figure 24-10.** Channel Suspend/Resume Operation




### 24.5.3.4. Event Input Actions

The event input actions are available only on the 4least significant DMA channels. For details on channels with event input support, refer to the in 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 Control B register (CHCTRLB.EVIE) must be written to '1'. Refer also to Events.

**Table 24-1.** Event Input Action

| Action | CHCTRLB.EVACT | CHCTRLB.TRGSRC |
|---|---|---|
| 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 | |

#### Normal Transfer

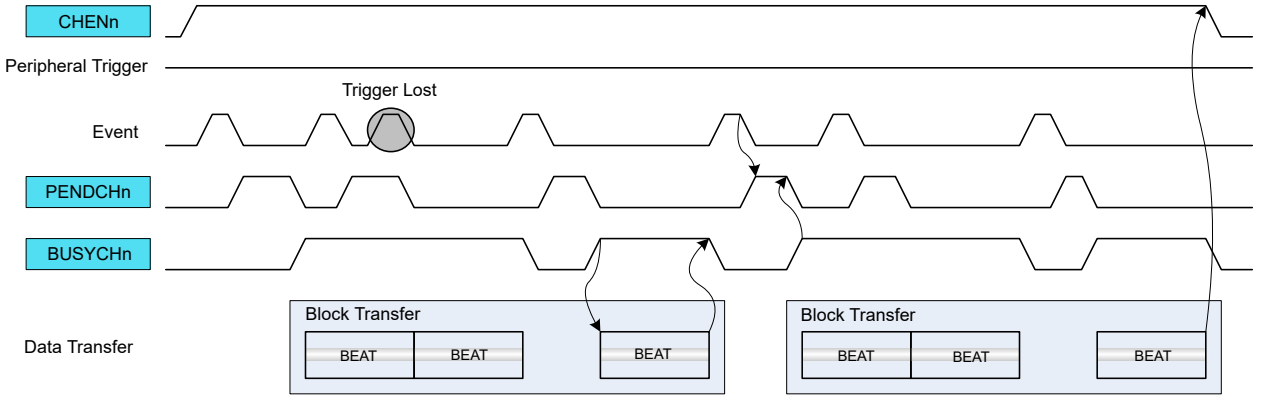
The event input is used to trigger a beat or burst transfer on peripherals.

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.


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

**Figure 24-11.** Beat Event Trigger Action



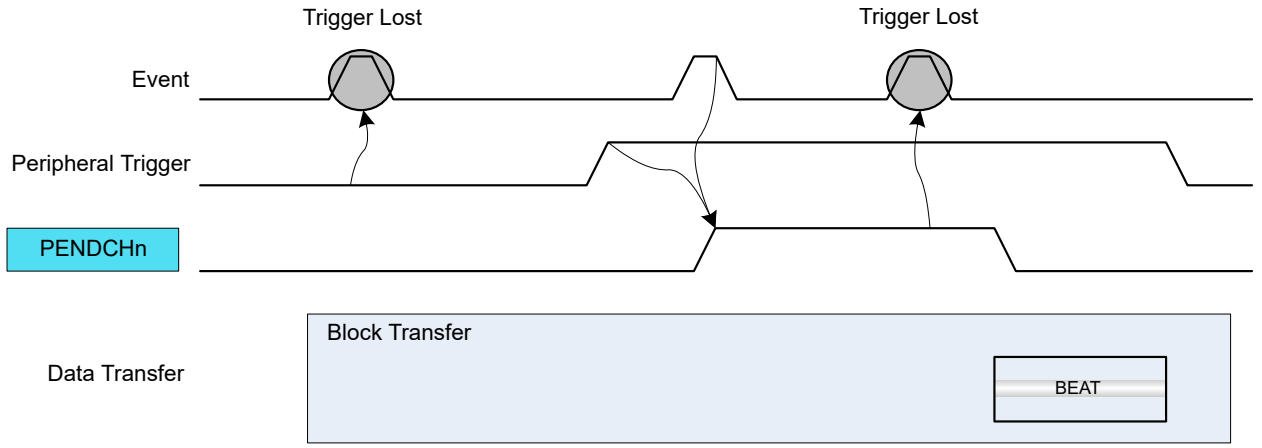

## Conditional Transfer on Strobe

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 24-12.** Periodic Event with Beat Peripheral Triggers




## Conditional Transfer

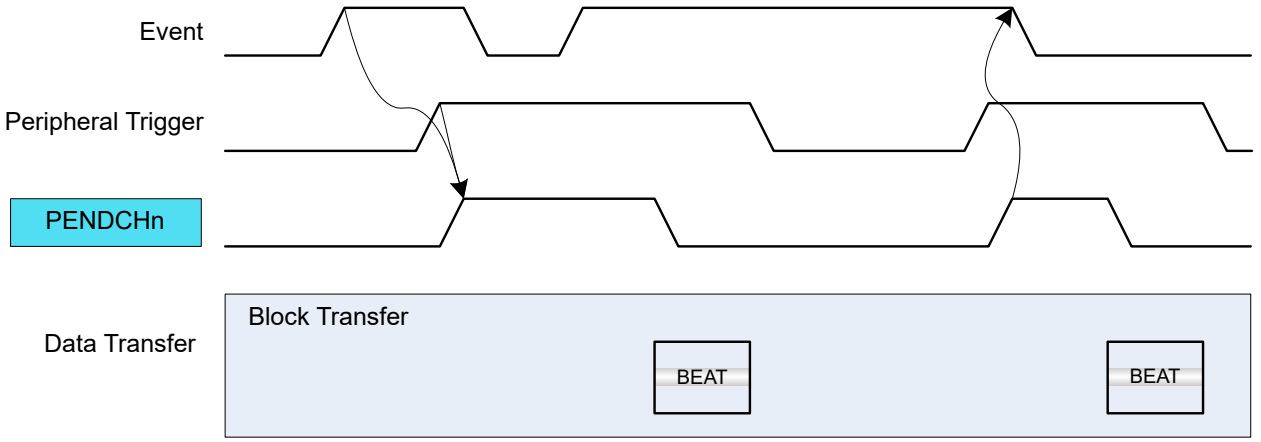
The event input is used to trigger a conditional transfer on peripherals with pending transfer requests. For example, this type of event can be used for peripheral-to-peripheral transfers, where one peripheral is the source of event and the second peripheral is the source of the trigger.

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.


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

The figure below shows an example where conditional event is enabled with peripheral beat trigger requests.

**Figure 24-13.** Conditional Event with Beat Peripheral Triggers



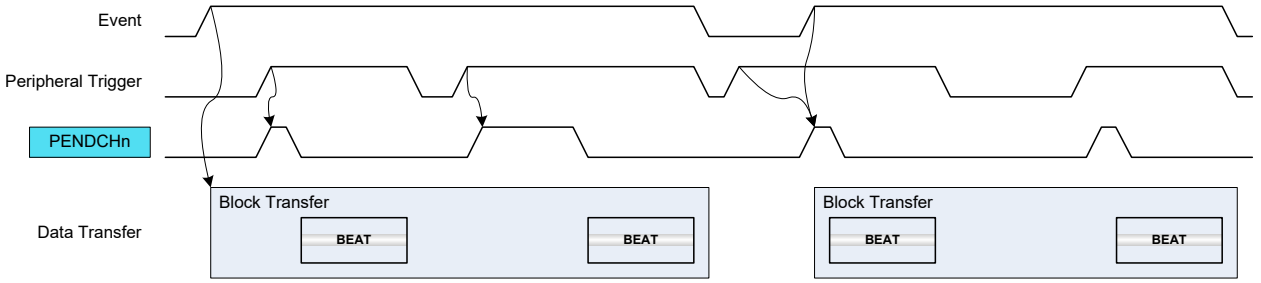

### Conditional Block Transfer

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 24-14.** Conditional Block Transfer with Beat 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.


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

#### 24.5.3.5. 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.

#### 24.5.3.6. 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: 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.

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.

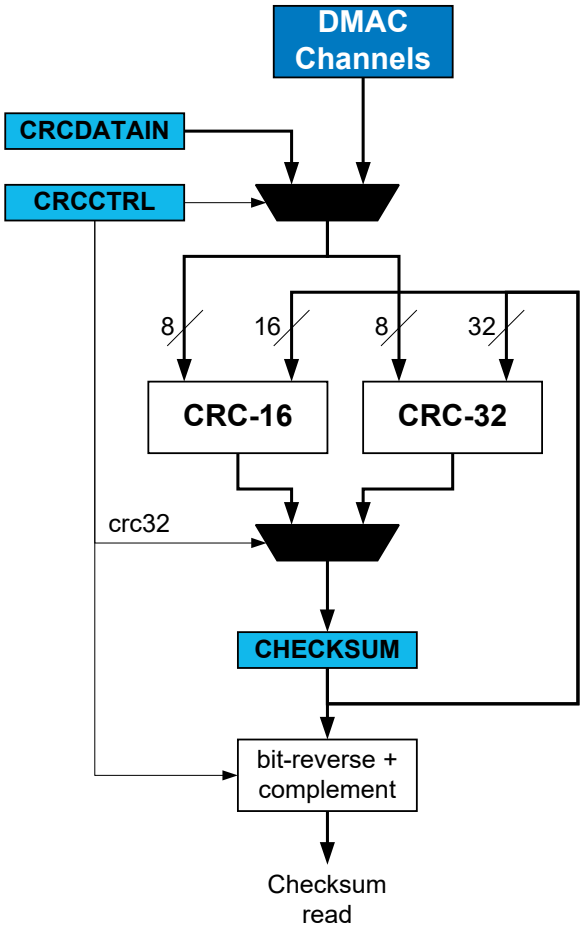
- CRC-16:
  - Polynomial: $x^{16}+ x^{12}+ x^5+ 1$
  - Hex value: 0x1021
- CRC-32:
  - Polynomial: $x^{32}+x^{26}+ x^{23}+ x^{22}+x^{16}+ x^{12}+ x^{11}+ x^{10}+ x^8+ x^7+ x^5+ x^4+ x^2+ x + 1$
  - Hex value: 0x04C11DB7

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 24-15.** 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.

### 24.5.4. 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.


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

- **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.

**Important:** Optionally 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 to read the Interrupt Pending register (INTPEND), which provides the lowest channel number with pending interrupt and the respective interrupt flags, for channels with ORed interrupt requests.

**Important:** Interrupts must be globally enabled for interrupt requests to be generated.

**Important:** Channels with dedicated interrupt lines do not refresh the INTPEND register.

### 24.5.5. 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


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

- 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.

### 24.5.6. Sleep Mode Operation

Each DMA channel can be configured to operate in any sleep mode. To be able to run in standby, the RUNSTDBY bit in Channel Control A register (CHCTRLA.RUNSTDBY) must be written to '1'. The DMAC can wake up the device using interrupts from any sleep mode or perform actions through the Event System.

For channels with CHCTRLA.RUNSTDBY = 0, it is up to software to stop DMA transfers on these channels and wait for completion before going to standby mode using the following sequence:

1. Suspend the DMAC channels for which CHCTRLA.RUNSTDBY = 0.
2. Check the SYNCBUSY bits of registers accessed by the DMAC channels being suspended.
3. Go to sleep.
4. When the device wakes up, resume the suspended channels.

**Note:** In Stand-by Sleep mode, the DMAC can only access RAM when it is not back biased (PM.STDBYCFG.BBIASxx = 0x0)

**Note:** In Stand-by Sleep mode, the DMAC can access the LP SRAM only when the power domain PD1 is not in retention and PM.STDBYCFG.BBIASLP=0x0. The DMAC can access the SRAM in Stand-by Sleep mode only when the power domain PD2 is not in retention and PM.STDBYCFG.BBIASHS = 0x0.

### 24.5.7. 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.


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

## 24.6. Peripheral Dependencies

**Table 24-2.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| DMAC | 0x44802000 | 42 : TCMPL0/ SUSP0/TERR0<br>43 : TCMPL1/ SUSP1/TERR1<br>44 : TCMPL2/ SUSP2/TERR2<br>45 : TCMPL3/ SUSP3/TERR3<br>46 : TCMPL4/ SUSP4/TERR4/ TCMPL5/ SUSP5/TERR5/ TCMPL6/ SUSP6/TERR6/ TCMPL7/ SUSP7/TERR7 | **AHB:**<br>CLKMSK[0]<br>MASK28<br>**APB:**<br>CLKMSK[0]<br>MASK29 | --- | 21<br>STATUS0<br>PERID21 | | Generators:<br>**CH0:**<br>CHANNEL[37]<br>**CH1:**<br>CHANNEL[38]<br>**CH2:**<br>CHANNEL[39]<br>**CH3:**<br>CHANNEL[40]<br>Users:<br>**CH0:**<br>USER[6],Resynch<br>**CH1:**<br>USER[7],Resynch<br>**CH2:**<br>USER[8],Resynch<br>**CH3:**<br>USER[9],Resynch | PD_CORE_SW |


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

## 24.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRL | 7:0 | | | | | | | DMAENABLE | SWRST |
| | | 15:8 | | | | | LVLENx3 | LVLENx2 | LVLENx1 | LVLENx0 |
| 0x02 | CRCCTRL | 7:0 | | | | | CRCPOLY[1:0] | | CRCBEATSIZE[1:0] | |
| | | 15:8 | | | 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 | | | | | | | CRCZERO | CRCBUSY |
| 0x0D | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| 0x0E | QOSCTRL | 7:0 | | | WRBQOS[1:0] | | FQOS[1:0] | | | |
| 0x0F | Reserved | | | | | | | | | |
| 0x10 | SWTRIGCTRL | 7:0 | SWTRIG7 | SWTRIG6 | SWTRIG5 | SWTRIG4 | SWTRIG3 | SWTRIG2 | SWTRIG1 | SWTRIG0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | PRICTRL0 | 7:0 | RRLVLEN0 | | | LVLPRI0[4:0] | | | | |
| | | 15:8 | RRLVLEN1 | | | LVLPRI1[4:0] | | | | |
| | | 23:16 | RRLVLEN2 | | | LVLPRI2[4:0] | | | | |
| | | 31:24 | RRLVLEN3 | | | LVLPRI3[4:0] | | | | |
| 0x18 ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | INTPEND | 7:0 | | | | | ID[3:0] | | | |
| | | 15:8 | PEND | BUSY | FERR | | | SUSP | TCMPL | TERR |
| 0x22 ... 0x23 | Reserved | | | | | | | | | |
| 0x24 | INTSTATUS | 7:0 | CHINT7 | CHINT6 | CHINT5 | CHINT4 | CHINT3 | CHINT2 | CHINT1 | CHINT0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | BUSYCH | 7:0 | BUSYCH7 | BUSYCH6 | BUSYCH5 | BUSYCH4 | BUSYCH3 | BUSYCH2 | BUSYCH1 | BUSYCH0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x2C | PENDCH | 7:0 | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 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] | | | | | | | |
| 0x3C ... 0x3E | 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x3F | CHID | 7:0 | | | | ID[4:0] | | | | |
| 0x40 | CHCTRLA | 7:0 | | RUNSTDBY | | | | | ENABLE | SWRST |
| 0x41 ... 0x43 | Reserved | | | | | | | | | |
| 0x44 | CHCTRLB | 7:0 | LVL[2:0] | | | EVOE | EVIE | EVACT[2:0] | | |
| | | 15:8 | TRIGSRC[7:0] | | | | | | | |
| | | 23:16 | TRIGACT[1:0] | | | | | | | |
| | | 31:24 | | | | | | | CMD[1:0] | |
| 0x48 ... 0x4B | Reserved | | | | | | | | | |
| 0x4C | CHINTENCLR | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x4D | CHINTENSET | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x4E | CHINTFLAG | 7:0 | | | | | | SUSP | TCMPL | TERR |
| 0x4F | CHSTATUS | 7:0 | | | | | | FERR | BUSY | PEND |


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

### 24.7.1. Control

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LVLENx3 | LVLENx2 | LVLENx1 | LVLENx0 |
| 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 both the DMAC and the CRC module are disabled (DMAENABLE and CRCENABLE are '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.

### 24.7.2. CRC Control

**Name:** CRCCTRL
**Offset:** 0x02
**Reset:** 0x0000
**Property:** PAC Write-Protection, Enable-Protected

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | CRCSRC[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CRCPOLY[1:0] | | CRCBEATSIZE[1:0] | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 13:8 – CRCSRC[5:0]** CRC Input Source
These bits select the input source for generating the CRC, as shown in the table below. 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 | NOACT | No action |
| 0x01 | IO | I/O interface |
| | CHN | DMA Channel n |

**Bits 3:2 – CRCPOLY[1:0]** CRC Polynomial Type
These bits define the size of the data transfer for each bus access when the CRC is used with I/O interface, as shown in the table below.

| 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.

### 24.7.3. 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.

### 24.7.4. CRC Checksum

**Name:** CRCCHKSUM
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write Protection, Enable-Protected

The CRCCHKSUM represents the 16-bit 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.

### 24.7.5. CRC Status

**Name:** CRCSTATUS
**Offset:** 0x0C
**Reset:** 0x00
**Property:** PAC Write Protection

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

**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.
When running CRC-32 and appending the checksum at the end of the packet (as little endian), the final checksum should be 0x2144df1c, and not zero. However, if the checksum is complemented before it is appended (as little endian) to the data, the final result in the checksum register will be zero. See the description of CRCCHKSUM to read out different versions of the checksum.

**Bit 0 – CRCBUSY** CRC Module Busy

This flag is cleared by writing a one to it when used with I/O interface. When used with a DMA channel, the bit is set when the corresponding DMA channel is enabled, and cleared when the corresponding DMA channel is disabled. This register bit cannot be cleared by the application when the CRC is used with a DMA channel.
This bit is set when a source configuration is selected and as long as the source is using the CRC module.


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

### 24.7.6. 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.

### 24.7.7. QOS Control

**Name:** QOSCTRL
**Offset:** 0x0E
**Reset:** 0x2A
**Property:** PAC Write Protection

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

**Bits 5:4 – WRBQOS[1:0]** Write-Back Quality of Service
These bits define the memory priority access during the write-back operation.

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

**Bits 5:4 – DQOS[1:0]** Data Transfer Quality of Service
These bits define the memory priority access during the data transfer operation.

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

**Bits 3:2 – FQOS[1:0]** Fetch Quality of Service
These bits define the memory priority access during the fetch 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.

### 24.7.8. Software Trigger Control

**Name:** SWTRIGCTRL
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** RW

| 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 |
|---|---|---|---|---|---|---|---|---|
| | SWTRIG7 | SWTRIG6 | SWTRIG5 | SWTRIG4 | SWTRIG3 | SWTRIG2 | SWTRIG1 | SWTRIG0 |
| Access | R/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 – SWTRIGn** Channel n Software Trigger
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.

### 24.7.9. Priority Control 0

**Name:** PRICTRL0
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** PAC Write Protection

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN3 | | | LVLPRI3[4:0] | | | | |
| Access | RW | | | RW | RW | RW | RW | RW |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN2 | | | LVLPRI2[4:0] | | | | |
| Access | RW | | | RW | RW | RW | RW | RW |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN1 | | | LVLPRI1[4:0] | | | | |
| Access | RW | | | RW | RW | RW | RW | RW |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RRLVLEN0 | | | LVLPRI0[4:0] | | | | |
| Access | RW | | | RW | RW | RW | RW | RW |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – RRLVLEN3** Level 3 Round-Robin Scheduling Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 3. For details on arbitration schemes, refer to *Arbitration*.

| Value | Description |
|---|---|
| 0 | Static arbitration scheme for channels with level 3 priority. |
| 1 | Round-robin arbitration scheme for channels with level 3 priority. |

**Bits 28:24 – LVLPRI3[4:0]** Level 3 Channel Priority Number
When round-robin arbitration is enabled (RRLVLEN3 = 1) for priority level 3, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 3.
When static arbitration is enabled (RRLVLEN3 = 0) for priority level 3, 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 (RRLVLEN3 written to '0').

**Bit 23 – RRLVLEN2** Level 2 Round-Robin Scheduling Enable
This bit controls which arbitration scheme is selected for DMA channels with priority level 2. For details on arbitration schemes, refer to *Arbitration*.

| Value | Description |
|---|---|
| 0 | Static arbitration scheme for channels with level 2 priority. |
| 1 | Round-robin arbitration scheme for channels with level 2 priority. |


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

**Bits 20:16 – LVLPRI2[4:0]** Level 2 Channel Priority Number

When round-robin arbitration is enabled (RRLVLEN2 = 1) for priority level 2, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 2.
When static arbitration is enabled (RRLVLEN2 = 0) for priority level 2, 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 (RRLVLEN2 written to '0').

**Bit 15 – RRLVLEN1** Level 1 Round-Robin Scheduling Enable

For details on arbitration schemes, refer to *Arbitration*.

| Value | Description |
|---|---|
| 0 | Static arbitration scheme for channels with level 1 priority. |
| 1 | Round-robin arbitration scheme for channels with level 1 priority. |

**Bits 12:8 – LVLPRI1[4:0]** Level 1 Channel Priority Number

When round-robin arbitration is enabled (RRLVLEN1 = 1) for priority level 1, this register holds the channel number of the last DMA channel being granted access as the active channel with priority level 1.
When static arbitration is enabled (RRLVLEN1 = 0) for priority level 1, 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 (RRLVLEN1 written to '0').

**Bit 7 – RRLVLEN0** Level 0 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 4:0 – LVLPRI0[4:0]** Level 0 Channel Priority Number

When round-robin arbitration is enabled (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 (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 (RRLVLEN0 written to '0').


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

### 24.7.10. Interrupt Pending

**Name:** INTPEND
**Offset:** 0x20
**Reset:** 0x0000
**Property:** -

This register allows the user to identify the lowest DMA channel with pending interrupt.

**Important:** INTPEND is dedicated to channels which are multiplexed on a single interrupt line (Channels 4 to 15).

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PEND | BUSY | FERR | | | SUSP | TCMPL | TERR |
| Access | R | R | R | | | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | ID[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 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 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 the Channel ID (ID) Suspend interrupt flag.

**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 the Channel ID (ID) Transfer Complete interrupt flag.

**Bit 8 – TERR** Transfer Error
This bit is read one when the channel selected by Channel ID field (ID) has pending Transfer Error interrupt.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Channel ID (ID) Transfer Error interrupt flag.

**Bits 3:0 – ID[3:0]** 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


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

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.

### 24.7.11. Interrupt Status

**Name:** INTSTATUS
**Offset:** 0x24
**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 |
|---|---|---|---|---|---|---|---|---|
| | 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 – CHINTn** Channel x Pending Interrupt
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.

### 24.7.12. Busy Channels

**Name:** BUSYCH
**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 |
|---|---|---|---|---|---|---|---|---|
| | BUSYCH7 | BUSYCH6 | BUSYCH5 | BUSYCH4 | BUSYCH3 | BUSYCH2 | BUSYCH1 | BUSYCH0 |
| Access | R/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 – BUSYCHn** Busy Channel x
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.

### 24.7.13. Pending Channels

**Name:** PENDCH
**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 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH | PENDCH |
| Access | R/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 – PENDCH** Pending Channel x
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.

### 24.7.14. Active Channel and Levels

**Name:** ACTIVE
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** R

| 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=0..3]
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.

### 24.7.15. Descriptor Memory Section Base Address

**Name:** BASEADDR
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** R/W

| 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.

### 24.7.16. Write-Back Memory Section Base Address

**Name:** WRBADDR
**Offset:** 0x38
**Reset:** 0x00000000
**Property:** RW

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[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 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[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 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[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 |
|---|---|---|---|---|---|---|---|---|
| | WRBADDR[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| 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.

### 24.7.17. Channel ID

**Name:** CHID
**Offset:** 0x3F
**Reset:** 0x00
**Property:** RW

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

**Bits 4:0 – ID[4:0]** Channel ID
These bits define the channel number that will be affected by the channel registers (CH*). Before reading or writing a channel register, the channel ID bit group must be written first.


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

### 24.7.18. Channel Control A

**Name:** CHCTRLA
**Offset:** 0x40
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

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

**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. |
| 1 | The reset operation is ongoing. |


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

### 24.7.19. Channel Control B

**Name:** CHCTRLB
**Offset:** 0x44
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CMD[1:0] | |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TRIGACT[1:0] | | | | | | | |
| Access | RW | RW | | | | | | |
| Reset | 0 | 0 | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TRIGSRC[7:0] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | LVL[2:0] | | | EVOE | EVIE | EVACT[2:0] | | |
| Access | RW | RW | RW | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 25:24 – CMD[1:0]** Software Command
These bits define the software commands. Refer to *Channel Suspend* and *Resume and Next Suspend Skip*.
These bits are not enable-protected.

| Value | Name | Description |
|---|---|---|
| 0x0 | NOACT | No action |
| 0x1 | SUSPEND | Channel suspend operation |
| 0x2 | RESUME | Channel resume operation |

**Bits 23:22 – TRIGACT[1:0]** Trigger Action
These bits define the trigger action used for a transfer.

| Value | Name | Description |
|---|---|---|
| 0x0 | BLOCK | One trigger required for each block transfer |
| 0x2 | BEAT | One trigger required for each beat transfer |
| 0x3 | TRANSACTION | One trigger required for each transaction |

**Bits 15:8 – TRIGSRC[7:0]** Trigger Source
These bits define the peripheral trigger which is source of the transfer. For additional information on trigger selection and trigger modes, refer to *Transfer Triggers and Actions* and CHCTRLB.TRIGACT.
Refer to the *Peripheral Dependencies* table for finding out the value corresponding to each module trigger source.

| Value | Name | Description |
|---|---|---|
| | PERIPHERAL | Peripheral trigger n selection |
| 0x00 | DISABLE | Only software/event triggers |


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

**Bits 7:5 – LVL[2:0]** Channel Arbitration Level

These bits define the arbitration level used for the DMA channel, where a high level has priority over a low level. For additional information on arbitration schemes, refer to the *Arbitration*.
These bits are not enable-protected.

| Value | Name | Description |
|---|---|---|
| 0x0 | LVL0 | Channel Priority Level 0 |
| 0x1 | LVL1 | Channel Priority Level 1 |
| 0x2 | LVL2 | Channel Priority Level 2 |
| 0x3 | LVL3 | Channel Priority Level 3 |

**Bit 4 – 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 descriptor Event Output Selection (BTCTRL.EVOSEL).
This bit is available only for the 4 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

| Value | Description |
|---|---|
| 0 | Channel event generation is disabled. |
| 1 | Channel event generation is enabled. |

**Bit 3 – EVIE** Channel Event Input Enable

This bit is available only for the 4 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

| 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 2:0 – EVACT[2:0]** Event Input Action

These bits define the event input action, as shown below. The action is executed only if the corresponding EVIE bit in the CHCTRLB register of the channel is set.
These bits are available only for the 4 least significant DMA channels. Refer to table: *User Multiplexer Selection* and *Event Generator Selection* of the Event System for details.

| 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 |


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

### 24.7.20. Channel Interrupt Enable Clear

**Name:** CHINTENCLR
**Offset:** 0x4C
**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.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

| 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.

### 24.7.21. Channel Interrupt Enable Set

**Name:** CHINTENSET
**Offset:** 0x4D
**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.
This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

| 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.

### 24.7.22. Channel Interrupt Flag Status and Clear

**Name:** CHINTFLAG
**Offset:** 0x4E
**Reset:** 0x00
**Property:** -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

| 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.

### 24.7.23. Channel Status

**Name:** CHSTATUS
**Offset:** 0x4F
**Reset:** 0x00
**Property:** -

This register affects the DMA channel that is selected in the Channel ID register (CHID.ID).

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | FERR | BUSY | PEND |
| Access | | | | | | R | R | R |
| Reset | | | | | | 0 | 0 | 0 |

**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.

## 24.8. DMA Descriptor Register Summary


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

### 24.8.1. Block Transfer Control

**Name:** BTCTRL
**Offset:** 0x00
**Reset:** 0x0000
**Property:** RW

The BTCTRL register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

**Table 24-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | STEPSIZE[2:0] | | | STEPSEL | DSTINC | SRCINC | BEATSIZE[1:0] | |
| Access | | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | BLOCKACT[1:0] | | EVOSEL[1:0] | | VALID |
| Access | | | | | | | | |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**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 + (BEATSIZE+1) * 1 |
| 0x1 | X2 | Next ADDR = ADDR + (BEATSIZE+1) * 2 |
| 0x2 | X4 | Next ADDR = ADDR + (BEATSIZE+1) * 4 |
| 0x3 | X8 | Next ADDR = ADDR + (BEATSIZE+1) * 8 |
| 0x4 | X16 | Next ADDR = ADDR + (BEATSIZE+1) * 16 |
| 0x5 | X32 | Next ADDR = ADDR + (BEATSIZE+1) * 32 |
| 0x6 | X64 | Next ADDR = ADDR + (BEATSIZE+1) * 64 |
| 0x7 | X128 | Next ADDR = ADDR + (BEATSIZE+1) * 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 |
|---|---|
| 0x0 | The Destination Address Increment is disabled |
| 0x1 | The Destination Address Increment is enabled |


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

**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

| Value | Description |
|---|---|
| 0x0 | The Source Address Increment is disabled |
| 0x1 | 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 |

**Bits 4:3 – BLOCKACT[1:0]** Block Action

These bits define what actions the DMAC should take after a block transfer has completed.

| Value | 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.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Event generation disabled |
| 0x1 | BLOCK | Event strobe when block transfer complete |
| 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 |
|---|---|
| 0x0 | The descriptor is not valid |
| 0x1 | The descriptor is valid |


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

### 24.8.2. Block Transfer Count

**Name:** BTCNT
**Offset:** 0x02
**Reset:** None
**Property:** RW

The BTCNT register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

**Table 24-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 24.8.3. Block Transfer Source Address

**Name:** SRCADDR
**Offset:** 0x04
**Reset:** None
**Property:** RW

The SRCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

**Table 24-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

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

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

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

**Bits 31:0 – SRCADDR[31:0]** Transfer Source Address
This bit group holds the source address corresponding to the last beat transfer address in the block transfer.


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

### 24.8.4. Block Transfer Destination Address

**Name:** DSTADDR
**Offset:** 0x08
**Reset:** None
**Property:** RW

The DSTADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

**Table 24-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

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

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

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

**Bits 31:0 – DSTADDR[31:0]** Transfer Destination Address
This bit group holds the destination address corresponding to the last beat transfer address in the block transfer.


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

### 24.8.5. Next Descriptor Address

**Name:** DESCADDR
**Offset:** 0x0C
**Reset:** None
**Property:** RW

The DESCADDR register offset is relative to (BASEADDR or WRBADDR) + Channel Number * 0x10

**Table 24-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

# 25. Supply Controller (SUPC)

## 25.1. Overview

The Supply Controller (SUPC) manages the voltage reference, power supply and supply monitoring of the device. It is also able to control multiple output pins named “SUPC_OUT[1:0]”.

The SUPC controls all the voltage regulators supplying the multiple 1.2v domain: switchable core domain (PD_CORE_SW), RAM power domain (PD_CORE_RAMx), backup (PD_CORE_BU) domain . It sets the voltage regulators according to the sleep modes, or the user configuration. It also controls a User LDO regulator to power an external component.

The SUPC controls several Brown-out Reset (BOR). BOR_VDDREG monitors the voltage applied to the regulators (VDDREG), BOR_VDDIO monitors the voltage applied to IOs. In Backup mode, a dedicated Duty Cycle BOR (DCBOR) is able to monitor both VDDREG and VDDIO/AVDD. BOR12 tightly coupled with the capless regulator guarantees the internal voltage of the core (VDDCORE).

The SUPC controls a configurable Low-Voltage Detector used to monitor VDDIO in both directions.

The SUPC controls an internal reference voltage and a current dependent on the temperature which can be used by analog modules like the ADC.

The SUPC also controls the device’s charge pumps.

## 25.2. Features

The SUPC controls the following analog supply elements:

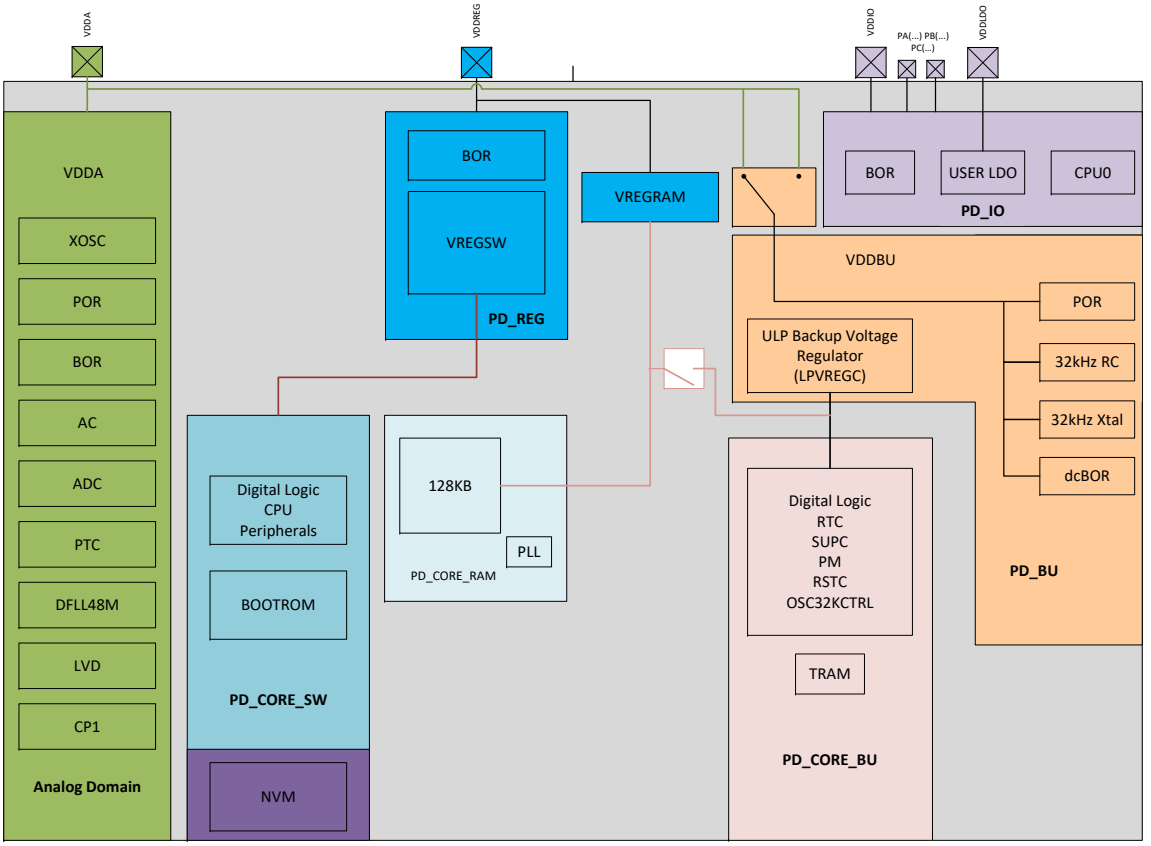
- Voltage Regulator System
  - Capless voltage regulators called VREGSW is used in Active, Idle, and Standby modes to supply PD_CORE_SW power domain.
  - Capless voltage regulator called VREGRAM used in Active, Idle, Standby, and Hibernate modes to supply VDDCORE_RAM domain and PLL.
  - User LDO regulator to power an external component.
- Voltage Reference System (Bandgap)
  - Reference voltage for ADC
  - Temperature sensor
  - Low-power mode in Standby Sleep mode
  - Charge Pump for I/O pad and analog cells, such as PTC/AC/ADC in case of low VDD voltage
- 3.3V Brown-out Reset (BOR) Detector
  - Two instances of BOR are used when calibrated; one to monitor VDDIO and one to monitor VDDREG power supply voltages, during Power-up, Active and Standby Sleep modes
  - Programmable threshold value loaded from USER CFG page at startup
  - Triggers resets
  - Operating modes: Continuous mode and Low-power mode in Standby and Hibernate modes
- 3.3V Low-Power Brown-out Reset (DCBOR)
  - Threshold values loaded from USER CFG page of CFM Flash Memory
  - Triggers resets
  - Operating modes: Continuous mode and Sampled mode (with programmable sampling frequency)
- 1.2V Brown-out Reset (BOR12)
  - Monitors VDDCORE power supply voltage


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

  - Tightly coupled with the capless regulator
  - Triggers resets
- 3.3V Programmable Low-Voltage Detector
  - Monitors VDDIO
  - Configurable threshold and direction
  - Can trigger Interrupt and generate EVSYS Events
- Output pins
  - Pin toggling on RTC event or by SUPC in Backup mode

## 25.3. Block Diagram

**Figure 25-1.** Supply Controller Block Diagram




Power domains (PDs) shown above are not independent. VDDCORE_*, VDDIO, and VDDREG share the same ground, GND. But AVDD has its own ground, AVSS. AVDD and VDDIO inputs must share the same supply, VDD.

The block diagram above also shows a variety of internally regulated power domains (VDDCORE PDs), nominally at 1.2V:

- **VDDCORE_BU:** Powers the backup domain. it contains peripherals that remain powered in the Backup Sleep mode
- **VDDCORE_SW:** The main voltage domain for the CPU, bus, and most peripherals, which can be switch off (hence the _SW suffix)
- **VDD_CORE_RAM:** This domain is used to retain the devices SRAM


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

The device's internal main voltage regulators have three different modes, controlled by the Supply Controller (SUPC):

- **Active Run Mode:** The default mode when the CPU and peripherals are running
- **Idle/Standby:** When the CPU and peripherals are in standby
- **Hibernate/Backup/Off Mode:** When the chip is in backup mode, the internal regulator is off, the VDDCORE_SW core power domain is OFF. The VDDCORE_BU backup domain is powered by the backup regulator (Low Power Voltage Regulator for Core - LPVREGC).

## 25.4. Signals Description

**Table 25-1.** Signals Description

| Signal Name | Type | Description |
|---|---|---|
| SUPC_OUT0 | Digital Output | SUPC Output 0 |
| SUPC_OUT1 | Digital Output | SUPC Output 1 |

## 25.5. Functional Description

### 25.5.1. Voltage Regulators

#### 25.5.1.1. Enabling, Disabling, and Resetting

The main regulators output supply level is automatically defined by the Sleep mode selected in the Power Manager module.

The user LDO regulator is disabled by default, as well as the additional Regulator(s)

The LDO regulator can be enabled and disabled by writing the ULDOEN bit high in the VREGCTRL register.

#### 25.5.1.2. VDDCORE Control

The VDDCORE supplies VDDCORE_SW, VDDCORE_RAM, VDDCORE_BU, if enabled in Standby mode.

#### 25.5.1.3. Charge Pump for Low VDDIO Voltage

It is highly recommended to activate charge pumps when VDDA/VDDIO is too low (<2.5V).
The control of Charge pump *n* (*n* = 0,1) is managed by setting the corresponding bit in VREGCTRL.CPEN[1:0]. When CPEN[*n*] bit is set, the enable and auto-enable modes of the charge pump[N] are set. In standby mode, the charge pumps are automatically turned OFF except if a consumer (PTC, ADC or AC) of the charge pump is requesting the charge pump.

### 25.5.2. Voltage References

The reference voltages are generated by a functional analog block controlled by the SUPC. It is providing a fixed-voltage source VREF for modules as ADC and AC as well as a temperature dependent voltage in support of a temperature sensor.

#### 25.5.2.1. Initialization

After reset, the voltage reference output is enabled and running at full power mode, and the temperature sensor is disabled.

#### 25.5.2.2. Enabling, Disabling, and Resetting

The temperature sensor is enabled/disabled by setting/clearing the Temperature Sensor Enable bit in the Voltage Reference register (VREFCTRL.TSEN).

### 25.5.3. Output and Input 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 SUPC_OUT*n* 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 the “Toggle Output Mode X is set to “RTCTGL”, the corresponding SUPC_OUT*n* pin will toggle when an RTC event occurs. Note that before configuring this mode, this is recommended to configure the RTC module properly. If the “Toggle Output Mode X is set to “BKUPTGL”, the corresponding SUPC_OUT*n* pin will toggle to one when the device goes to Backup Mode.

The BKIN register reflects the current state of the of the corresponding BKOUT pin if enabled (SUPC.BKOUT.EN = 1).

### 25.5.4. Brown-out Reset (BOR)

#### 25.5.4.1. Initialization

- BOR_VDDREG: After power-up or user reset, the BOR monitoring VDDREG is active. Its hysteresis values are loaded from NVM calibration User Config (UCFG) and stored in the BRCFGUSMOR_BOR register.
- BOR_VDDIO: After power-up or user reset, the BOR monitoring VDDIO is active. Its threshold and hysteresis values are loaded from NVM User Config (UCFG) and stored in the BRCFGUSMOR_BOR register.
- DCBOR: After power-up or user reset, the DCBOR used in Backup mode is inactive. Its threshold and hysteresis values are loaded from NVM Calibration User Config (UCFG) and stored in the BRCFGUDSSMOR_BOR register.

#### 25.5.4.2. 3.3V Regular Brown-out Reset

In all modes except the Backup mode the BOR_VDDREG and BOR_VDDIO/VDDA compare respectively the VDDREG and VDDIO voltage with the VDDREG and VDDIO brown-out threshold level. This level is set during power-up when NVM calibration values are loaded. When VDDREG or VDDIO crosses below the brown-out threshold level, BOR generates a Reset.

#### 25.5.4.3. 3.3V Duty Cycle Brown-out Reset (DCBOR)

To reduce power consumption, DCBOR is used in sampling mode. The Sampling Mode is a Low-Power mode where the DCBOR is being repeatedly enabled on a sampling clock’s ticks. The DCBOR will monitor the supply voltage for a short period of time and then go to a low-power disabled state until the next sampling clock tick. The frequency of the clock ticks is controlled by the Prescaler Select bit groups in the BOR register (BOR.DCBORPSEL). Refer to the “BOR – BOR Control” SFR for more details.

#### 25.5.4.4. BOR Filtering

For BOR generation we have an input filter to filter-out any glitches which are less than 100 nS. Supplementary BOR filtering is also possible for BOR_VDDIO and BOR_VDDREG by using the BOR.BORFILT bitfield.

### 25.5.5. Programmable Low Voltage Detector

The programmable LVD is able to monitor the VDDIO input power supply.

Before enabling the LVD by setting LVD.ENABLE bit, the threshold level bits field (LVD.LEVEL) and the direction (LVD.DIR) have to be configured. The LVD register is Enable-Protected, meaning that it can only be written when the LVD is disabled (LVD.ENABLE). As long as the Enable bits is ‘1’, any writes to LVD register will be discarded and a bus error will be generated. The Enable bits are not Enable-Protected.

#### 25.5.5.1. Settling Time

After being enabled, a start-up delay is required before the result of the comparison is ready. During the startup-time, the output value is zero (STATUS.LVDET).


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

After the start-up time has passed, the LVDET ready bit is set (STATUS.LVDRDY) indicating that STATUS.LVDET bit is available, and the appropriate LVD interrupt flag or output event are also generated. New comparisons are performed continuously until LVD.ENABLE bit is written to zero. The start-up time applies only to the first comparison. If LVD is disabled (LVD.ENABLE is written to 0), the STATUS.LVDRDY bit will take time to fall because of resynchronization. As a consequence, this is forbidden to re-enable immediately the LVD if the STATUS.LVDRDY field is still one.

#### 25.5.5.2. LVD Interrupt and Event

• **LVD Output Interrupt source:** LVD Output interrupt source status (STATUS.LVDET) is set to one when the VDDIO voltage value crosses the programmed threshold level (LVD.LEVEL) in the right direction (LVD.DIR). It is reset to zero when the VDDIO voltage value crosses the threshold level in the wrong direction (LVD.DIR). Note that if the LVD is just enabled and the voltage is already below the threshold (if DIR=1) or the voltage is already above the threshold (if DIR=0), then the LVD output interrupt is also generated.

• **LVD Output event:** The LVD can generate output event when the VDDIO voltage value crosses the programmed threshold level (LVD.LEVEL) in the right direction (LVD.DIR). To enable this feature, the LVD.OEVEN bit has to be set to one. Refer to the EVSYS Event System for details on configuring the event system.

#### 25.5.5.3. LVD Behavior in Sleep Mode

LVD is off in hibernate and backup sleep modes.

LVD is automatically turned OFF in standby mode except if the LVD.RUNSTDBY bit is set.

### 25.5.6. Interrupts

The SUPC has the following interrupt sources, which are either synchronous or asynchronous wake-up sources:

- LVD Ready (LVDRDY), synchronous
- LVD Detection (LVDDET), asynchronous
- ULDORDY, 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 via INTENSET. 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 SUPC 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.

### 25.5.7. Power Management

The SUPC can operate in all sleep modes including backup sleep mode.

### 25.5.8. 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.

### 25.5.9. Register Access Protection

Registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC). Here is the list of the write-protected registers:


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

- INTENCLR
- INTENSET
- BOR
- LVD
- VREGCTRL
- VREFCTRL
- BBPS
- BKOUT

## 25.6. Peripheral Dependencies

**Table 25-2.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| SUPC | 0x45002000 | 3 : LVDDET/ LVDRDY/ ULDORDY | **APB:** CLKMSK[0] MASK4 | --- | 4 STATUS0 PERID4 | | Generators: **LVDET:** CHANNEL[1] | PD_CORE_SW |


© 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 | INTENCLR | 7:0 | | | | | | ULDORDY | LVDRDY | LVDET |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | INTENSET | 7:0 | | | | | | ULDORDY | LVDRDY | LVDET |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | INTFLAG | 7:0 | | | | | | ULDORDY | LVDRDY | LVDET |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | STATUS | 7:0 | | | | | | ULDORDY | LVDRDY | LVDET |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | SYNCBUSY | 7:0 | | | | | | | | BOR |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | BOR | 7:0 | | DCBORPSEL[2:0] | | | | | | |
| | | 15:8 | | | | | | | BORFILT[1:0] | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | LVD | 7:0 | | | | RUNSTDBY | OEVEN | DIR | ENABLE | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | LEVEL[3:0] | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | VREGCTRL | 7:0 | | | LVHIB | LVSTDBY | | OFFSTDBY | VREGOUT[1:0] | |
| | | 15:8 | ULDOLEVEL[1:0] | | ULDOSTDBY | ULDOEN | CPEN[3:0] | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | BKUP_VLD | SRAM_VLD | | | | | | |
| 0x20 | VREFCTRL | 7:0 | | | | TSEN | | | LPHIB | LPSTDBY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 ... 0x27 | Reserved | | | | | | | | | |
| 0x28 | BKOUT | 7:0 | | | | | | | EN1 | EN0 |
| | | 15:8 | | | | | | | CLR1 | CLR0 |
| | | 23:16 | | | | | | | SET1 | SET0 |
| | | 31:24 | | | | | TGLOM1[1:0] | | TGLOM0[1:0] | |
| 0x2C | BKIN | 7:0 | | | | | | | BKIN[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 25.7.1. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** RW

**Table 25-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ULDORDY | LVDRDY | LVDET |
| Access | | | | | | R/W | RW | RW |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – ULDORDY** User LDO Regulator Ready Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit disables the ULDORDY interrupt.
This bit will read as the current value of the ULDORDY interrupt enable.

**Bit 1 – LVDRDY** Low Voltage Detector Ready Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit disables the LVDRDY interrupt.
This bit will read as the current value of the LVDRDY interrupt enable.

**Bit 0 – LVDET** Low Voltage Detector Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit disables the LVDET interrupt.
This bit will read as the current value of the LVDET interrupt enable.


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

### 25.7.2. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** RW

**Table 25-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ULDORDY | LVDRDY | LVDET |
| Access | | | | | | R/W | RW | RW |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – ULDORDY** User LDO Regulator Ready Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit enables the ULDORDY interrupt.
This bit will read as the current value of the ULDORDY interrupt enable.

**Bit 1 – LVDRDY** Low Voltage Detector Ready Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit enables the LVDRDY interrupt.
This bit will read as the current value of the LVDRDY interrupt enable.

**Bit 0 – LVDET** Low Voltage Detector Interrupt Enable
Writing a zero to this bit has no effect.
Writing a one to this bit enables the LVDET interrupt.
This bit will read as the current value of the LVDET interrupt enable.


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

### 25.7.3. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x008
**Reset:** 0x00000000
**Property:** RW

**Note:** Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

**Table 25-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ULDORDY | LVDRDY | LVDET |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – ULDORDY** User LDO Regulator Ready Interrupt
Set to one if ULDO is ready meaning that output voltage is correct.
Set by hardware when the User LDO is ready to operate.
Write one to clear the corresponding bit.

**Bit 1 – LVDRDY** Low Voltage Detector Ready Interrupt
Set to one if LVD is ready to operate.
Set by hardware the Low Voltage Detector is ready to operate.
Write one to clear the corresponding bit.

**Bit 0 – LVDET** Low Voltage Detector Interrupt
Set to one if VDDIO crosses the threshold voltage in the good direction according to LVD.DIR.
Set by hardware if VDDIO crosses the threshold voltage in a “good” direction according to LVD.DIR.
Write one to clear the corresponding bit.


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

### 25.7.4. Flag status

**Name:** STATUS
**Offset:** 0x00C
**Reset:** 0x00000000
**Property:** R

**Table 25-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | ULDORDY | LVDRDY | LVDET |
| Access | | | | | | R | R | R |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – ULDORDY** User LDO Regulator Status
It corresponds to vreg_ready_mv signal from ULDO.
Set by hardware when the User LDO is ready to operate.

**Bit 1 – LVDRDY** Low Voltage Detector Ready Status
Set by hardware when the Low Voltage Detector is ready.

**Bit 0 – LVDET** Low Voltage Detector Status
Set to one if VDDIO crosses the threshold voltage in the “good” direction according to LVD.DIR.


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

### 25.7.5. Synchronisation Busy

**Name:** SYNCBUSY
**Offset:** 0x010
**Reset:** 0x00000000
**Property:** R

**Table 25-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | BOR |
| Access | | | | | | | | R |
| Reset | | | | | | | | 0 |

**Bit 0 – BOR** BOR Synchronization Busy
When hardware sets this bit, no writes are permitted to the BOR register. After writing to the BOR register wait for this flag to clear before making additional writes.


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

### 25.7.6. BOR Control

**Name:** BOR
**Offset:** 0x014
**Reset:** 0x00000000
**Property:** RW

**Table 25-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BORFILT[1:0] | |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

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

**Bits 9:8 – BORFILT[1:0]** BOR filtering

| Value | Name | Description |
|---|---|---|
| 0x0 | NOFILT | No digital filtering |
| 0x1 | FILT32US | 32us filtering |
| 0x2 | FILT125US | 125us filtering |
| 0x3 | FILT250US | 250us filtering |

**Bits 6:4 – DCBORPSEL[2:0]** Duty Cycle BOR Prescaler Select

| Value | Name | Description |
|---|---|---|
| 0x0 | NODIV | Not Divided |
| 0x1 | DIV2 | Divide clock by 2 |
| 0x2 | DIV4 | Divide clock by 4 |
| 0x3 | DIV8 | Divide clock by 8 |
| 0x4 | DIV16 | Divide clock by 16 |
| 0x5 | DIV32 | Divide clock by 32 |
| 0x6 | DIV64 | Divide clock by 64 |
| 0x7 | DIV128 | Divide clock by 128 |


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

### 25.7.7. LVD Control

**Name:** LVD
**Offset:** 0x018
**Reset:** 0x00000000
**Property:** RW

**Table 25-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LEVEL[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| 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 | OEVEN | DIR | ENABLE | |
| Access | | | | RW | RW | RW | RW | |
| Reset | | | | 0 | 0 | 0 | 0 | |

**Bits 19:16 – LEVEL[3:0]** Threshold Level.

| Value | Description |
|---|---|
| 1111 | 3.630 V |
| 1110 | 3.330 V |
| 1101 | 3.000 V |
| 1100 | 2.927 V |
| 1011 | 2.824 V |
| 1010 | 2.727 V |
| 1001 | 2.500 V |
| 1000 | 2.400 V |
| 0111 | 2.308 V |
| 0110 | 2.243 V |
| 0101 | 2.202 V |
| 0100 | 2.124 V |
| 0011 | 2.017 V |
| 0010 | 1.890 V |
| 0001 | 1.805 V |
| 0000 | 1.727 V |

**Bit 4 – RUNSTDBY** Run during Standby


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

| Value | Description |
|---|---|
| 0 | Run during standby disabled |
| 1 | Run during standby enabled |

**Bit 3 – OEVEN** Output Event Enable

| Value | Description |
|---|---|
| 0 | Output events disabled |
| 1 | Output events enabled |

**Bit 2 – DIR** Direction

| Value | Name | Description |
|---|---|---|
| 0x0 | rising | rising detection |
| 0x1 | falling | falling detection |

**Bit 1 – ENABLE** Enable

| Value | Description |
|---|---|
| 0 | Low Voltage Detection disabled |
| 1 | Low Voltage Detection enabled |


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

### 25.7.8. VREG Control

**Name:** VREGCTRL
**Offset:** 0x01C
**Reset:** 0x00000004
**Property:** RW

**Note:** During normal operation, all voltage regulators that are in use must be left in the On state to allow for the proper transition between different low-power/standby states.

**Table 25-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | BKUP_VLD | SRAM_VLD | | | | | | |
| Access | RW | RW | | | | | | |
| Reset | 0 | 0 | | | | | | |

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | ULDOLEVEL[1:0] | | ULDOSTDBY | ULDOEN | CPEN[3:0] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | LVHIB | LVSTDBY | | OFFSTDBY | VREGOUT[1:0] | |
| Access | | | RW | RW | | RW | RW | RW |
| Reset | | | 0 | 0 | | 1 | 0 | 0 |

**Bit 31 – BKUP_VLD** Backup Domain Valid Status Bit
**Note:** Hardware Cleared by Reset and Power Management Unit Whenever Backup power domain is lost. Software (SW) Set by Backup Domain Initialization Code in BOOT ROM when completed.

| Value | Description |
|---|---|
| 0 | Backup “BKUP” Power Domain has encountered a power loss and contents are not valid.(DEFAULT) |
| 1 | Backup “BKUP” Power Domain has not encountered a power loss. Contents are valid. |

**Bit 30 – SRAM_VLD** SRAM VALID Status Bit
**Notes:**

1. Hardware Clear by Reset and Power Management Units whenever SRAM power domain is lost.
2. Should be set by Users SW Code once valid data has been written into System SRAM when " MCRAMC.CTRLA.ENABLE = 0" after SRAM is initialized by user.

| Value | Description |
|---|---|
| 0 | SRAM has encountered a power loss and contents are not valid. (DEAFAULT) |
| 1 | SRAM has not encountered a power loss. |


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

**Bits 15:14 – ULDOLEVEL[1:0]** User LDO Voltage Level Selection

| Value | Name | Description |
|---|---|---|
| 0x0 | 1p2v | Vout = 1.2v |
| 0x1 | 1p5v | Vout = 1.5v |
| 0x2 | 1p8v | Vout = 1.8v |
| 0x3 | 2p5v | Vout = 2.5v |

**Bit 13 – ULDOSTDBY** User LDO Voltage Regulator Configuration

| Value | Name | Description |
|---|---|---|
| 0x0 | OFFINSTDBY | Regulator is OFF while in sleep mode equal or deeper than standby mode. It is OFF in hibernate and backup mode as well. |
| 0x1 | ONINSTDBY | Regulator is ON in Standby mode. is OFF from Hibernate mode. It is OFF in backup mode as well. |

**Bit 12 – ULDOEN** User LDO Voltage Regulator Enable

| Value | Description |
|---|---|
| 0x0 | User LDO is disabled |
| 0x1 | User LDO is enabled |

**Bits 11:8 – CPEN[3:0]** Charge Pump Enable and Auto-enable

| Value | Description | Requirements |
|---|---|---|
| 0x0 | All charge pumps disabled. | AVDD ≥ 2.5v |
| 0x1 | Enable charge pump for I/O analog mux and Analog Comparator (AC) | AVDD < 2.5v |
| --- | Reserved | |
| 0x3 | Enable charge pumps for I/O, AC, ADC | |

**Note:**

1. When AVDD < 2.5v the corresponding appropriate CPEN must be enabled.

**Bit 5 – LVHIB** Low Voltage Hibernate Enable

| Value | Name | Description |
|---|---|---|
| 0x0 | 1p2v | In Hibernate mode, VDDCORE_BU and VDDCORE_RAM are set to 1.2v. |
| 0x1 | 0p8v | In Hibernate mode, VDDCORE_BU and VDDCORE_RAM are set to 0.8v. |

**Bit 4 – LVSTDBY** Low Voltage Standby Enable

| Value | Name | Description |
|---|---|---|
| 0x0 | 1p2v | In standby mode, VDDCORE_BU, VDDCORE_RAM, VDDCORE_SW and optionally VDDCOREUSB/PLL are set to 1.2v. |
| 0x2 | 0p8v | In standby mode, VDDCORE_BU, VDDCORE_RAM, VDDCORE_SW and optionally VDDCOREUSB/PLL are set to 0.8v. |

**Bit 2 – OFFSTDBY** Off in Standby Control for VREGSW[N-1]

| Value | Name | Description |
|---|---|---|
| 0x0 | OFF | In standby mode, VREGSWx are OFF. |
| 0x1 | ON | In standby mode, VREGSWx are ON. |

**Bits 1:0 – VREGOUT[1:0]** VREG Output Control in RUN mode only
Enable by production fuse by CALSUPC.VREGOUTEN.

| Value | Name | Description |
|---|---|---|
| 0x0 | 1p2v | In Active mode, VDDCORE_RAM, VDDCORE_BU, and VDDCORE_SW are set to 1.2v |
| 0x1 | 1p0v | In Active mode, VDDCORE_RAM, VDDCORE_BU, and VDDCORE_SW are set to 1.0v. |
| 0x2 | 0p8v | In Active mode, VDDCORE_RAM, VDDCORE_BU, and VDDCORE_SW are set 0.8v. |


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

### 25.7.9. VREF Control

**Name:** VREFCTRL
**Offset:** 0x020
**Reset:** 0x00000002
**Property:** RW

**Table 25-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | TSEN | | | LPHIB | LPSTDBY |
| Access | | | | RW | | | RW | RW |
| Reset | | | | 0 | | | 1 | 0 |

**Bit 4 – TSEN** Temperature Sensor Output Enable

| Value | Description |
|---|---|
| 0 | Temperature sensor output to ADC disabled |
| 1 | Temperature sensor output to ADC Enabled |

**Bit 1 – LPHIB** Bandgap and Regulators Low Power Hibernate Enable

| Value | Name | Description |
|---|---|---|
| 0x0 | FullPower | In hibernate mode, bandgap is set to nominal power mode. As a consequence, enabled regulator(s) are set to nominal power mode. |
| 0x1 | LowPower | In hibernate mode, bandgap is set to low power mode. As a consequence, enabled regulator(s) are set to low power mode. |

**Bit 0 – LPSTDBY** Bandgap and Regulators Low Power Standby Enable

| Value | Name | Description |
|---|---|---|
| 0x0 | FullPower | In standby mode, bandgap is set to nominal power mode. As a consequence, enabled regulator(s) are set to nominal power mode. |
| 0x1 | LowPower | In standby mode, bandgap is set to low power mode. As a consequence, enabled regulator(s) are set to low power mode. |


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

### 25.7.10. Backup Output Control

**Name:** BKOUT
**Offset:** 0x028
**Reset:** 0x00000000
**Property:** RW

**Note:** These register bits are useful for indicating to users external application logic MCU Backup mode entry/exit.

**Table 25-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | TGLOM1[1:0] | | TGLOM0[1:0] | |
| Access | | | | | RW | RW | RW | RW |
| Reset | | | | | 0 | 0 | 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 | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bits 24:25, 26:27 – TGLOM** Toggle Output Mode

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | The output does not toggle. |
| 0x1 | RTCTGL | The output toggles on RTC event. |
| 0x2 | BKUPTGL | The output is set when the device enters backup mode or battery backup mode. The output should then be cleared by software. |

**Bits 16, 17 – SET** Set Output

| Description |
|---|
| Writing a zero to this bit has no effect. |
| Writing a one to this bit will set the corresponding output. |

**Bits 8, 9 – CLR** Clear Output
Reading this bit returns '0'

| Description |
|---|
| Writing a zero to this bit has no effect. |


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

| Description |
| --- |
| Writing a one to this bit will set the corresponding output. |

**Bits 0, 1 – EN** Enable Output


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

### 25.7.11. Backup Input Control

**Name:** BKIN
**Offset:** 0x02C
**Reset:** 0x00000000
**Property:** R

**Table 25-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BKIN[1:0] | |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bits 1:0 – BKIN[1:0]** Backup Input Value

| Description |
|---|
| These bits are cleared when the corresponding backup I/O pin detects a logical low level on the input pin. |
| These bits are set when the corresponding backup I/O pin detects a logical high level on the input pin. |


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

# 26. Power Manager (PM)

## 26.1. 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 (SYSRST is not enabled in sleep mode) can restore the device from a Sleep mode to Active mode.

Performance level technique consists of adjusting the regulator output voltage to reduce power consumption. The user can select on the fly the performance level configuration which best suits the application.

The application configures which power domains are on and off during Standby, Backup, and Hibernate sleep modes. When the application selects to enter a sleep mode, hardware automatically enters that mode when it detects no activity. Based on activity monitoring, power domain gating is managed automatically by hardware without software intervention. This technique is transparent for the application while minimizing the static consumption.

In Hibernate and Backup mode, the PM allows retaining the state of the I/O lines, preventing I/O lines from toggling during wake-up.

## 26.2. Features

The Power management module controls:

- Sleep modes: Idle, Standby, Hibernate, Backup, and Off
- Two performance levels
- SleepWalking available in standby mode.
- I/O lines retention in hibernate and Backup mode

## 26.3. 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.

## 26.4. Interrupts

The interrupt request line is connected to the interrupt controller. Using the PM interrupt requires the interrupt controller to be configured first.

## 26.5. Debug Operation

When the CPU is halted in Debug mode, the PM continues normal operation. If the 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.

Hot plugging in standby mode is supported.

Hot plugging in Hibernate or backup mode is not supported as the DSU module is not powered.


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

Cold plugging in Hibernate or backup mode is supported if the external reset duration is superior to the corresponding sleep mode wakeup time, see *Electrical Characteristic* chapter for additional information.

## 26.6. Register Access Protection

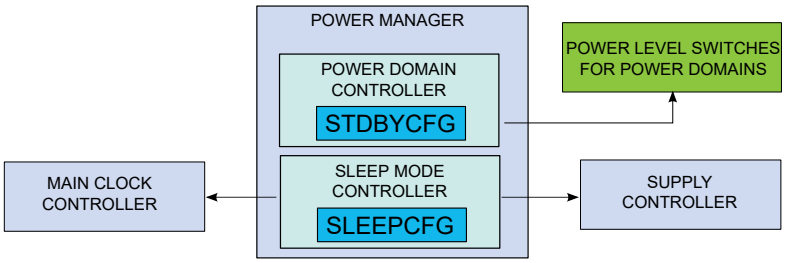
Registers with write-access can be write-protected optionally by the Peripheral Access Controller (PAC), except for the following:

- Interrupt Flag Status and Clear register (INTFLAG)

Optional PAC Write-Protection is denoted by the "PAC Write-Protection" property in each individual register description.

## 26.7. Block Diagram

**Figure 26-1.** Power Manager Block Diagram




## 26.8. Functional Description

### 26.8.1. Power Domains

Leaving aside the domains supplied by external power pins, such as VDDREG, AVDD, VDDIO , the device is split into these internal power domains:

- VDDCORE_BU_PD
- VDDCORE_SW_PD
- VDDCORE_RAM_PD

VDDCORE_SW_PD: is the Switchable Power Domain. It contains the CPU and all the peripherals except those located in the backup power domain.

VDDCORE_BU_PD: Contains the backup peripherals: OSC32KCTRL, SUPC, RSTC, RTC, TRAM, and the PM itself.

VDDCORE_RAM_PD: Contains the system RAM.

### 26.8.2. 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:

- **Active mode:** CPU is running, all power domains are active.
- **Idle sleep mode:** The CPU is stopped. Synchronous clocks are stopped except when requested. The logic state is retained.
- **Standby sleep mode:** The CPU is stopped as well as the peripherals. The logic state is retained..
- **Hibernate sleep mode:** PD_CORE_SW 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 external pins). The The PD_CORE_RAMx power domain can be retained according to software configuration.


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

- **Backup sleep mode:** Only the backup domain (VDD_BU_PD)is kept powered to allow few features to run (RTC, 32 kHz clock sources, and wake-up from external pins).
- **Off sleep mode:** The entire device is powered off.

### 26.8.3. 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.

### 26.8.4. Basic Operation

#### 26.8.4.1. Initialization

After a Power-on Reset, the PM is enabled, the device is in Active mode, and all the power domains are in active state.

#### 26.8.4.2. Enabling, Disabling, and Resetting

The PM is always enabled and cannot be reset.

#### 26.8.4.3. 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[2:0]) select the level of the sleep mode.

**Note:** Due to clock domain synchronization a small latency occurs between the store instruction and actual writing of the SLEEPCFG.SLEEPMODE[2:0]. Software must ensure that the SLEEPCFG register reads the desired value of SLEEPMODE before issuing the WFI instruction.

Entry and exit from the various sleep modes is shown in the following table:

**Table 26-1.** Sleep Mode Entry and Exit

| Mode | Mode Entry | Wake-Up Sources |
|---|---|---|
| IDLE | SLEEPCFG.SLEEPMODE = IDLE + WFI | Synchronous [2] (APB, AHB), asynchronous [1] |
| STANDBY | SLEEPCFG.SLEEPMODE = STANDBY + WFI | Synchronous[3], asynchronous [1] |
| HIBERNATE | SLEEPCFG.SLEEPMODE = HIBERNATE + WFI | Hibernate reset detected by the RSTC |
| BACKUP | SLEEPCFG.SLEEPMODE = BACKUP + WFI | Backup reset detected by the RSTC |
| OFF | SLEEPCFG.SLEEPMODE = OFF + WFI | 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 clock activity, regulators, and NVM state are described in the sections below. Refer to the *Power Domain Controller* section for the power domain gating effect.


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

**Table 26-2.** Modes of Operation

| Mode | Main Clock | CPU | AHBx & APBx Clocks | GCLK Clocks | Oscillators ONDEMAND= | | Regulators | | | NVM |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | 0 | 1 | VREGSW | VREGRAM | LPVREG | |
| ACTIVE | Run | Run | Run | RiR(2) | Run | RiR(2) | On | On | Off | Active |
| IDLE | Run | Stop | Stop(1) | RiR(2) | Run | RiR(2) | On | On | Off | Active |
| STANDBY | Stop | Stop | Stop(1) | Stop(1) | RiRR(3) | RiR(2) | On | LPM(4) | Off | LPM(4) |
| HIBERNATE | Stop | Stop | Stop | Stop | Stop | Stop | Off | LPM(4) | Off | Off |
| BACKUP | Stop | Stop | Stop | Stop | Stop | Stop | Off | Off | On | Off |
| OFF | Stop | Stop | Off | Off | Off | Off | Off | Off | Off | Off |

**Notes:**

1. Stop except if running during Sleepwalking mode.
2. RiR = Run if Requested.
3. RiRR = Run if Requested or RUNSTDBY = 1.
4. LPM = Low-Power Mode.
5. On or Off, controlled by VREGCTRL.AVREGEN.

#### 26.8.4.3.1. Idle Mode

The 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. Before entering the Idle mode, the user must select the Idle Sleep mode in the Sleep Configuration register (SLEEPCFG.SLEEPMODE = IDLE).

Additionally, if the SLEEPONEXIT bit in the Arm Cortex System Control register (SCR) is set, the Idle mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine. See Arm Cortex documentation for details). This mechanism can be useful for applications that only require the processor to run when an interrupt occurs.

**Exiting Idle mode**: The processor wakes the system up when it detects any non-masked interrupt with sufficient priority to cause exception entry according to the PRIMASK register. The system goes back to the Active mode. The CPU and affected modules are restarted.

GCLK clocks, regulators, and RAM are not affected by the Idle Sleep mode and operate in normal mode.

#### 26.8.4.3.2. 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.CTRLA.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).

Additionally, if the SLEEPONEXIT bit in the Arm Cortex System Control register (SCR) is set, the Standby mode will be entered when the CPU exits the lowest priority ISR (Interrupt Service Routine.


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

This mechanism can be useful for applications that only require the processor to run when an interrupt occurs. Refer to the "Arm Cortex documentation" for details.

**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 awakened, 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. In Sleepwalking, a peripheral involved in the Sleepwalking task can wake up the system whatever the interrupt type (asynchronous or synchronous).

**Note:** While sleep-walking, a peripheral involved in the sleepwalking task can wake up the system regardless of the interrupt type (asynchronous or synchronous).

#### 26.8.4.3.3. Hibernate and Backup Modes

Hibernate and Backup modes allow the lowest power consumption aside from off. The device is entirely powered off except for the VDDCORE_BU_PD domain. In Hibernate mode, the VDDCORE_RAM_PD power domain can be retained according to software configuration. 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 reset except the CTRLA.IORET bit.

**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 (SLEEPMODE = HIBERNATE or = BACKUP).

**Exiting Hibernate or Backup mode**: This mode is triggered when a Hibernate or Backup Reset is detected by the Reset Controller (RSTC).

**Notes:**

1. In Hibernate mode, the VREGRAM (in Low-Power mode) regulator is used to allow powering the VDDCORE_RAM_PD power domain which is fully retained.
2. In Backup mode, the Backup Regulator (LPVREGC) is used.
3. Before entering Backup Sleep mode, it is recommended to poll the INTFLAG.SLEEPRDY bit to make sure that the backup regulator is ready. Let us say that the WFI instruction is executed whereas this flag is not set, then the system will go in a “pseudo backup mode where the VDDCORE_SW_PD power domain is turned off, but the VREGRAM is still used. Then the system will really go to Backup Sleep mode once the INTFLAG.SLEEPRDY field is set.
4. When WFI instruction is executed, before entering Hibernate or Backup, the PM waits until all the synchronous clocks have been properly turned off by MCLK and MCLK honors all clock requests from peripherals before turning off the synchronous clocks. Once there are no clock requests pending, the VDDCORE_SW_PD power domain can be turned off. Consequently, software is responsible to disable peripherals before executing the sleep instruction to avoid pending clock requests.
5. Let us say an interrupt occurs at the moment when the CPU is sleeping and MCLK is honoring a pending clock request then the device will wake up and the CPU can service the corresponding interrupt handler. The RCAUSE.BACKUP bit is zero as no backup reset has occurred.
6. To reduce further the consumption, the SUPC->VREGCTRL.LVHIB bit can be set to reduce the voltage level of the regulator. This sleep mode is just an extension of the HIBERNATE mode, called LV-HIBERNATE. Please refer to VREGCTRL register of the SUPC for details.

#### 26.8.4.3.4. 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 after it is verified that the SLEEPMODE field has been set to Off mode.


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

**Exiting Off mode**: This mode is left by pulling the RESET pin low.

#### 26.8.4.4. I/O Line State Retention in Hibernate and Backup Modes

When entering Hibernate or Backup mode, PORTs are 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 CTRLA register (CTRLA.IORET).

If IORET = 0 when exiting Hibernate or Backup mode, the I/O lines configuration is released and driven by the reset values of each 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 reprogrammed each PORT.

#### 26.8.4.5. Power Domain Controller

| Power Domain State | | | | | |
|---|---|---|---|---|---|
| Sleep Mode | STDBYCFG.RAMCFG | HIBCFG.RAMCFG | PD_CORE_SW | PD_CORE_BU | PD_CORE_RAMx |
| 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 or LVStandby - case 1 (see Note1) | RET | N/A | active | active | all retained |
| Standby or LVStandby - case 2 (see Note1) | OFF | N/A | active | active | off |
| Hibernate or LVHibernate - case 1 (see Note1) | N/A | RET | off | active | all retained |
| Hibernate or LVHibernate - case 2 (see Note1) | N/A | off | off | active | off |
| Backup | N/A | off | off | active | off |
| off | N/A | N/A | off | off | off |

**Note:** (Note1): LowVoltage option for LV_Standby or LV-Hibernate does not have any effect on this power-domain state table.

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 running a Sleepwalking task, VDDCORE_RAM_PD power domain is active whatever the STDBYCFG.RAMCFG is set to retain all RAM memory.

**Hibernate mode:**

When entering a Hibernate mode, the VDDCORE_SW_PD power domain is off. As in Standby mode, the VDDCORE_RAM_PD power domain can be selectively turned ON or OFF by using HIBCFG.RAMCFG field.

When entering Backup mode, the VDDCORE_SW_PD and VDDCORE_RAM_PD power domains are off, as well as the dedicated power domain (VDDCORE_USB_PD). VDDCORE_BU_PD is still active (powered by LPVREGC).

When entering Backup mode, the VDDCORE_SW_PD and VDDCORE_RAM_PD power domains are off, as well as the dedicated power domain. VDDCORE_BU_PD is still active (powered by LPVREGC).

**OFF mode:**

When entering Off mode, all the power domains are off. I/Os are in high-impedance mode except the RESET_N pad which is still in input mode able to detect a reset to wake up the chip.


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

### 26.8.4.6. Regulators, RAMs, and NVM State in Sleep Modes

**Idle sleep mode:**

GCLK clocks, regulators, NVM, and RAM are not affected in Idle sleep mode and operate normally.

**Standby sleep mode:**

By default, in Standby sleep mode, the RAMs, NVM, and regulators are automatically set in Low-Power mode in order to reduce power consumption. If SUPC.VREGCTRL.LVSTDBY bit is zero, then the output voltage regulator is kept to nominal value (1.2V). If LVSTDBY is enabled, this mode is called LVSTANDBY.

- The RAM can be disabled when STDBYCFG.RAMCFG = 0 when the device is in Standby mode. RAM contents are retained.
- If not powered-down, the RAM can be set in Low-Power mode according to STDBYCFG.LPRAM.
- If neither powered-down nor Low-Power mode is enabled, the RAM data is retained and not set in retention mode. In this case, the wakeup-time is better as there is no need to turn OFF/ON the RAM power-switches.

**Non-Volatile Memory:**

The NVM is automatically set in low power mode according to Flash Panel configuration.

NVM is automatically set in Low-Power mode according to Flash Panel configuration (FCR.CTRLB.SLP[1:0]). Flash can be in configured to enter auto-standby (with fast wakeup) or flash-hibernate (powered down w/ slow wakeup).

CTRLB.SLP[1:0]: NVM Power Reduction Mode selection during System Standby Sleep

- 11 = Enter Flash Hibernate on entry to Standby w/ wakeup on first access
- 10 = Enter Flash Hibernate on entry to Standby w/ wakeup to Auto Standby
- 01 = RSVD
- 00 = Enter Auto Standby on entry to Standby

Regulators: by default, in standby sleep mode, the PM analyzes the device activity to set the regulators (for PD_CORE_SW and PD_CORE_RAMx) to either low or full power mode, to supply the VDDCORE_SW and VDDCORE_RAM, according to the SUPC->VREGCTRL.LVSTDBY bit.

**Hibernate sleep mode:**

- The RAM can be powered-down according to HIBCFG.RAMCFG bit when the device is in standby mode. If powered-down, the data will be lost (VDDCE and VDDPE are powered-down).
- If not powered-down, the RAM can be set in low-power mode according to the HIBCFG.LPRAM bit. if in lowpower mode, the data is retained (RET1N signal is set, VDDPE voltage is powered-down, VDDCE is kept powered).
- If neither powered-down nor low-power mode, the RAM are retained and not set in retention mode (RET1N is inactive and VDDPE/VDDCE are kept powered-on). In this case, the wakeup-time is better as there is no need to turn OFF/ON the VDDPE/VDDCE power-switch.
- "Non-Volatile Memory : The NVM is powered-down.
- "Regulators: VDDCORE_RAM_PD is ON with reduced voltage if SUPC->VREGCTRL.LVHIB is set. VREGSW is OFF.
  **Note:** if LVHIB bit is zero, then the VREGRAM output voltage is not reduced. On the contrary, if LVHIB is set this mode is called LVHibernate sleep mode.

**Backup sleep mode:**

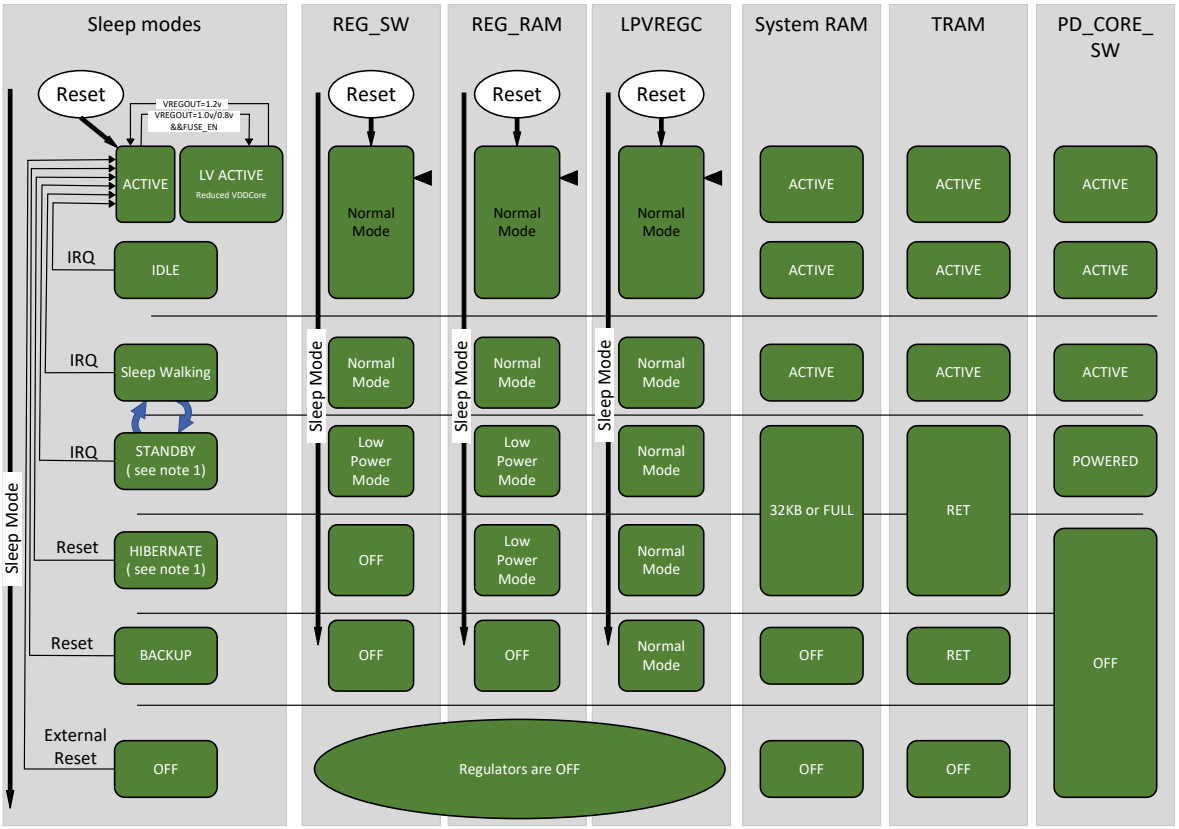
- The RAMs located in VDDCORE_RAM_PD are powered down.
- NVM is powered down.


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

- VDDCORE_RAM_PD is OFF as well as VDDCORE_SW_PD.
- The backup domain VDDCORE_BU_PD is powered by LPVREG.

### 26.8.4.7. Power Mode Definition and Power State Table

**Figure 26-2.** OPERATING CONDITIONS AND SLEEPWALKING




**Note:** In standby mode, if SUPC->VREGCTRL.LVSTDBY bit is one, then the regulators output voltage are reduced, and this mode is called LVStandby mode.

**Note:** In standby mode, if SUPC->VREGCTRL.LVHIB bit is one, then the regulators output voltage are reduced, and this mode is called LVHibernate mode.

Here is a list of supported Mode to Mode Transition. This list is just a simplified overview of the possible transition and might be incomplete.

- ACTIVE -> LV_ACTIVE -> ACTIVE (if CALSUPC.VREGOUTEN is set)
- ACTIVE -> Standby (LV optional) -> ACTIVE
- ACTIVE -> Hibernate (LV optional) -> ACTIVE
- ACTIVE -> Backup -> ACTIVE
- ACTIVE -> OFF -> ACTIVE


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

**Table 26-3.** POWER MODE DEFINITION

| Power Mode Name | Definition Address | Possible next states | NOTES |
|---|---|---|---|
| Active | All clocks running , VDDCORE is nominal | ACTIVE -> Idle, LV Active, Standby, LV Standby, Hibernate, LV Hibernate, Backup, OFF | CPU at max frequency in Run Sleepwalking only gates CPU clock |
| Idle | Some clock gating | IDLE -> ACTIVE | CPU at max frequency in Run Sleepwalking only gates CPU clock |
| Sleepwalking | CPU clock gating | Sleepwalk -> Standby , Sleepwalk -> LV standby Sleepwalk -> ACTIVE | Sleepwalking only gates CPU clock |
| LV Active | Reduced Vddcore | ACTIVE | User SW controlled for VREGCTRL voltage and clock frequency..(Fuse enabled) Exit to full Run must wait for VREGS to stabilize before WAIT instruction. Max CPU freq less then 8MHZ |
| Standby | Everything is clock gated | ACTIVE, Sleepwalk | Peripheral clocks are on demand |
| LV Standby | Reduced Vddcore Everything is clock gated BG in LP Duty Cycle Mode | ACTIVE, Sleepwalk | Longer delay to exit to ACTIVE or Sleep-walk state |
| Hibernate | Only Backup domain and SYS RAM powered | ACTIVE | |
| LV Hibernate | Reduced Vddcore Only Backup domain and SYS RAM powered BG in LP Duty Cycle Mode | ACTIVE | |
| Backup | Only backup domain is powered | ACTIVE | |
| Off | Only pads are powered | ACTIVE | Only wakeup is from RESET pin |

## 26.8.5. Advanced Features

### 26.8.5.1. 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 awakened by an interrupt (from a peripheral involved in SleepWalking) or enter again into Standby Sleep mode.

In Standby when Sleepwalking is ongoing:

- All the power domains are turned ON including PDRAM power domain
- Low-power mode of the regulators is not activated during sleepwalking

### 26.8.5.2. Wake-Up Time

As shown in the following figure, total wake-up time depends on:

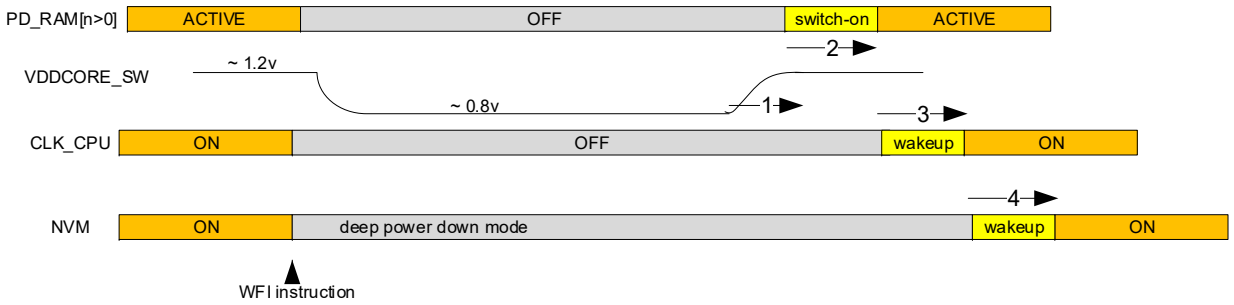
- Latency due to Reference and Regulator effect.
  As example, if the device is in LV standby sleep mode using the voltage regulators for VDDCORE_SW_PD (VREGSW*n*, *n*=0,1) in low power mode, the voltage level is lower than the one used in active mode. When the device wakes up, it takes a certain amount of time for the main regulators (VREGSW*n* and VREGRAM) to transition to the voltage level corresponding to active mode, causing additional wake-up time.
- 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.
- Latency due to the CPU clock source wake-up time.


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

- Latency due to the NVM wake-up time from deep power down mode.

**Figure 26-3.** Total Wake-Up Time from Standby Sleep Mode

1: Latency due to reference wakeup (low-power mode to full-power mode) and regulator up voltage scaling.
2: Latency due to VDDCORE_RAM_PD switching ON.
3: Latency due to clock source wakeup time.
4: Latency due to NVM wakeup time from deep power down mode.




### 26.8.5.3. Interrupts

Sleep Mode Entry Ready (SLEEPRDY): indicates that the device is ready to enter backup sleep mode, that is, it indicates that the low power bandgap of the LPVREGC regulator is ready, given the fact that it can take several 100's of microseconds at power-up for the low power bandgap to settle.

This interrupt is a synchronous wake-up source. For example, application can setup the device to enter backup sleep mode, go to standby sleep and get a wakeup interrupt from SLEEPRDY so it can then go to Backup Sleep.

This 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. This 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 in the NVIC module. 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 pe-ripheral 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.

### 26.8.5.4. Sleep Mode Operation

The Power Manager is always active in sleep modes.

## 26.9. Peripheral Dependencies

**Table 26-4.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| PM | 0x45000000 | 2 : SLEEPRDY | **APB:** CLKMSK[0] MASK3 | --- | 3 STATUS0 PERID3 | | | PD_CORE_SW |


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

## 26.10. 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<br>...<br>0x03 | Reserved | | | | | | | | | |
| 0x04 | INTENCLR | 7:0 | | | | | | | | SLEEPRDY |
| 0x05 | INTENSET | 7:0 | | | | | | | | SLEEPRDY |
| 0x06 | INTFLAG | 7:0 | | | | | | | | SLEEPRDY |
| 0x07 | Reserved | | | | | | | | | |
| 0x08 | STDBYCFG | 7:0 | | | | | | LPRAM | | |
| 0x09 | HIBCFG | 7:0 | | | | | | LPRAM | | |


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

### 26.10.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** RW

**Table 26-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | IORET | | |
| Access | | | | | | RW | | |
| Reset | | | | | | 0 | | |

**Bit 2 – IORET** I/O Retention

When entering Hibernate or Backup mode, I/O ports are powered off but the pin configurations are retained. When the device exits either mode, pin configurations are either released or stretched, based on this bit:

| Value | Name | Description |
|---|---|---|
| 0x0 | NOIORET | When the device exits the HIBERNATE or BACKUP mode, the I/O line configuration are released. |
| 0x1 | IORET | When the device exits the HIBERNATE or BACKUP mode, the I/O line configuration are stretched. |


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

### 26.10.2. Sleep Configuration

**Name:** SLEEPCFG
**Offset:** 0x01
**Reset:** 0x02
**Property:** RW

**Table 26-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bits 2:0 – SLEEPMODE[2:0]** Sleep Mode

| Value | Definition |
|---|---|
| 0x0 | Reserved |
| 0x1 | Reserved |
| 0x2 | CPU, AHB and APB clocks are OFF (IDLE) |
| 0x3 | Reserved |
| 0x4 | All Clocks are OFF (STANDBY) |
| 0x5 | Backup domain is ON as well as some PDRAMs (HIBERNATE) |
| 0x6 | Only Backup domain is powered ON (BACKUP) |
| 0x7 | All power domains are powered OFF (OFF) |


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

### 26.10.3. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x04
**Reset:** 0x00
**Property:** RW

**Table 26-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 0 – SLEEPRDY** Backup Sleep Mode Entry Ready Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will disable the SLEEPRDY interrupt.
**Note:** Reading this bit returns whether this interrupt is enabled (1 = enabled).


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

### 26.10.4. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x05
**Reset:** 0x00
**Property:** RW

**Table 26-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 0 – SLEEPRDY** Backup Sleep Mode Entry Ready Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the SLEEPRDY interrupt.
**Note:** Reading this bit returns whether this interrupt is enabled (1 = enabled).


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

### 26.10.5. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x06
**Reset:** 0x00
**Property:** RW

**Note:** Subsequent to an interrupt flag being cleared, the flag must be read back to verify the clear before exiting the ISR. Failure to do this can result in duplicate interrupts.

**Table 26-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SLEEPRDY |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – SLEEPRDY** Backup Sleep Mode Entry Ready
**Note:** Writing a '1' to this bit will clear the flag and the interrupt.

| Value | Description |
|---|---|
| 0 | Device is not ready to enter Backup Sleep Mode. |
| 1 | Device is ready to enter Backup Sleep Mode. |


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

### 26.10.6. Standby Configuration

**Name:** STDBYCFG
**Offset:** 0x08
**Reset:** 0x04
**Property:** RW

**Table 26-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 2 – LPRAM** Low Power RAM Enable

| Value | Description |
|---|---|
| 0x0 | RAM is not retained in standby when Low Power mode is active |
| 0x1 | RAM is retained in standby when Low Power mode is active |


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

### 26.10.7. Hibernate Configuration

**Name:** HIBCFG
**Offset:** 0x09
**Reset:** 0x04
**Property:** RW

**Table 26-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 2 – LPRAM** Low Power RAM Enable

| Value | Description |
|---|---|
| 0x0 | RAM is not retained in standby when Low Power mode is active |
| 0x1 | RAM is retained in standby when Low Power mode is active |


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

# 27. Reset Controller (RSTC)

## 27.1. Overview

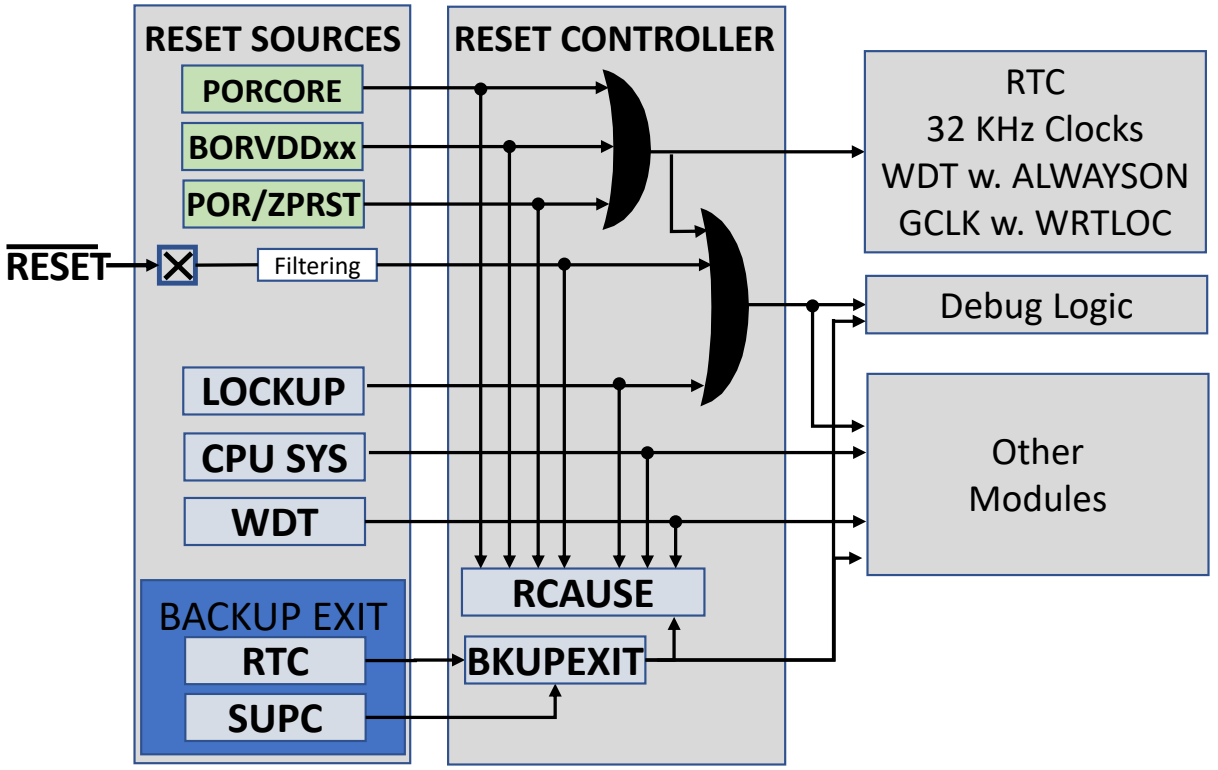
The Reset Controller (RSTC) manages the reset of the Microcontroller. It issues a Microcontroller reset, sets the device to its initial state and allows the reset source to be identified by software.

## 27.2. 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, PORCORE, BORVDDx
  - **User reset sources:** External Reset (RESET), Watchdog Reset, System Reset Request, CPU Lockup Reset
  - **Backup exit sources:** Real-Time Counter (RTC)

## 27.3. Block Diagram

**Figure 27-1.** Reset Controller Block Diagram




## 27.4. Signals Description

**Table 27-1.** Signals Description

| Signal Name | Type | Description |
|---|---|---|
| $\overline{\text{RESET}}$ | Digital input | External reset |


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

## 27.5. Functional Description

The Reset Controller collects the various Reset sources and generates Resets for the device.

### 27.5.1. Basic Operation

#### 27.5.1.1. Initialization

After a Power-on Reset, the RSTC is enabled and the Reset Cause (RCAUSE) register indicates the Reset source.

#### 27.5.1.2. Enabling, Disabling, and Resetting

The RSTC module is always enabled.

#### 27.5.1.3. Reset Causes and Effects

The latest Reset cause is available in the RCAUSE register and can be read during the application boot sequence to determine proper action.

These are the groups of Reset sources:

- **Power supply Reset:** Resets caused by an electrical issue. It covers POR and PORCORE resets.
- **Brown-out Reset:** Resets caused by BORVDDx detectors.
- **User Reset:** Resets caused by the application. It covers external Resets, system Reset requests, CPU Lockup Reset and Watchdog Timer Reset.
- **Backup Reset:** Reset cause when leaving Backup or Hibernate mode due to wake event RTC, or $\overline{\text{RESET}}$.

The following table lists the parts of the device that are reset, depending on the Reset type.

**Table 27-2.** Effects of Reset Causes

| Effect:<br>Reset: | CPU Logic,<br>Bus Logic | RTC(1), OSC32KCTRL, BKOUT, BOR(2), SUPC, VREGCTRL & VREFCTRL(3) registers,<br>CTRLA bits of PM | Flash Panel Controller | Debug Logic | Other Modules |
|---|---|---|---|---|---|
| POR, PORCORE | Reset | Reset | Reset | Reset | Reset |
| External Reset<br>BORVDDx | Reset | No Reset | No Reset | Reset | Reset |
| WDT Reset<br>Lockup Reset<br>System Reset Request | Reset | No Reset | No Reset | No Reset | Reset |
| RTC | - | No Reset | Reset | Reset | - |

**Notes:**

1. RTC.DBGCTRL is reset in Backup Sleep mode.
2. The BOR register is retained during Backup mode and is not reset when exiting.

The external Reset is generated when pulling the RESET pin low.

The POR, PORCORE, and BORVDDx (for example, BORVDDREG) Reset sources are generated by the analog modules inside the PWR_SMOR or PWR_SMOR_DS controlled by the Supply Controller Interface (SUPC).

The WDT Reset is generated by the Watchdog Timer (WDT).

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 additional information, refer to the "Arm® Cortex® Technical Reference Manual" which is available for download at www.arm.com).


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

The Lockup Reset is a Reset generated by the CPU when it enters a lockup state. For additional information, refer to the "Arm® Cortex® Technical Reference Manual" which is available for download at www.arm.com).

### 27.5.2. Power Management

The Reset Controller module is always on.

### 27.5.3. Sleep Mode Operation

The RSTC module is active in all sleep modes.

### 27.5.4. Debug Operation

When the CPU is halted in Debug mode, the RSTC continues normal operation.

If the DBGCTRL.LCKUPDIS bit is set, the CPU LOCKUP reset source is disabled in Debug mode. Refer to the DBGCTRL register for additional information.

### 27.5.5. PAC Write-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.

When the CPU is halted in Debug mode, all write-protection is automatically disabled. Write-protection does not apply for accesses through an external debugger.

## 27.6. Peripheral Dependencies

**Table 27-3.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| RSTC | 0x45004000 | | **APB:**<br>CLKMSK[0]<br>MASK5 | --- | 5<br>STATUS0<br>PERID5 | | | PD_CORE_SW |


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

## 27.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | RCAUSE | 7:0 | SYST | WDT | EXT | BORVDDIO | | BORVDDREG | PORCORE | POR |
| | | 15:8 | | | | | | | LOCKUP | BACKUP |
| 0x02 | BKUPEXIT | 7:0 | HIB | | | | | | RTC | |
| 0x03 | Reserved | | | | | | | | | |
| 0x04 | DBGCTRL | 7:0 | | | | | | | | LCKUPDIS |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 27.7.1. Reset Cause

**Name:** RCAUSE
**Offset:** 0x0
**Reset:** None
**Property:** R

**Note:** Only one bit is set at any one time. This register only reflects the last event.

**Table 27-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SYST | WDT | EXT | BORVDDIO | | BORVDDREG | PORCORE | POR |
| Access | R | R | R | R | | R | R | R |
| Reset | | | | | | | | |

**Bit 9 – LOCKUP**
CPU has entered lockup state. (for details, refer to the "Arm® Cortex™ Technical Reference Manual" available at www.arm.com).

**Bit 8 – BACKUP**
Reset cause when leaving Backup or Hibernate mode due to wake event (RTC) or external reset.

**Bit 7 – SYST**
The CPU was reset because a software reset (system reset) from main CPU has been performed.
Reset generated by a System Reset Request by the CPU by asserting the SYSRESETREQ bit located in the Reset Control register of the CPU (for details, refer to the "Arm® Cortex™ Technical Reference Manual" available at www.arm.com).

**Bit 6 – WDT**
Reset generated by Watchdog Timer.

**Bit 5 – EXT**
The CPU was reset due to the RESET pin being asserted.
Reset by the $\overline{\text{RESET}}$ pin driven low.

**Bit 4 – BORVDDIO**
The CPU was reset due to the VDDIO supply voltage being lower than the brown-out threshold level 3.3 V.
Reset generated by VDDIO BOR Detector.

**Bit 2 – BORVDDREG**
The CPU was reset due to the VDDREG supply voltage being lower than the brown-out threshold level 3.3 V.
Reset by VDDREG BOR Detector.


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

**Bit 1 – PORCORE**

The CPU was reset due to the supply voltage being lower than the brown-out threshold level 1.2 V.
Reset by Core Power-on Reset.

**Bit 0 – POR** Power-on Reset

The CPU was reset due to the supply voltage being lower than the power-on level
Reset generated by Power-on Reset.


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

### 27.7.2. Backup Exit Source

**Name:** BKUPEXIT
**Offset:** 0x2
**Reset:** 0x00
**Property:** R

Implemented only if RSTC_BACKUP_IMPLEMENTED=1.
**Note:** Backup exit only: RTC=1
Hibernate exit: HIB and RTC = 1

**Table 27-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 7 – HIB** Exit from Hibernate Sleep Mode
Wake up from Hibernate Sleep Mode.

**Bit 1 – RTC** RTC Backup Reset Condition
Wake up from Real Time Counter (RTC) interrupt.


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

### 27.7.3. Debug Control

**Name:** DBGCTRL
**Offset:** 0x4
**Reset:** 0x00000000
**Property:** RW

**Table 27-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | LCKUPDIS |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

**Bit 0 – LCKUPDIS** Lockup Disable
If set the CPU LOCKUP Reset source is disabled in debug mode.
(The Lockup Reset is a Reset generated by the CPU when it enters a lockup state (for details refer to the ARM® Cortex™ Technical Reference Manual on www.arm.com)).


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

# 28. External Interrupt Controller (EIC)

## 28.1. 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, on falling, on both edges, or on high or low levels. Each external pin has a configurable filter to remove spikes. Each external pin can also be configured to be asynchronous in order to wake up the device from sleep modes where all clocks have been disabled. External pins can also generate an event. Each external pin can be defined as secured or non-secured, where secured pins can only be handled by secure accesses.

A separate non-maskable interrupt (NMI) is also 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.

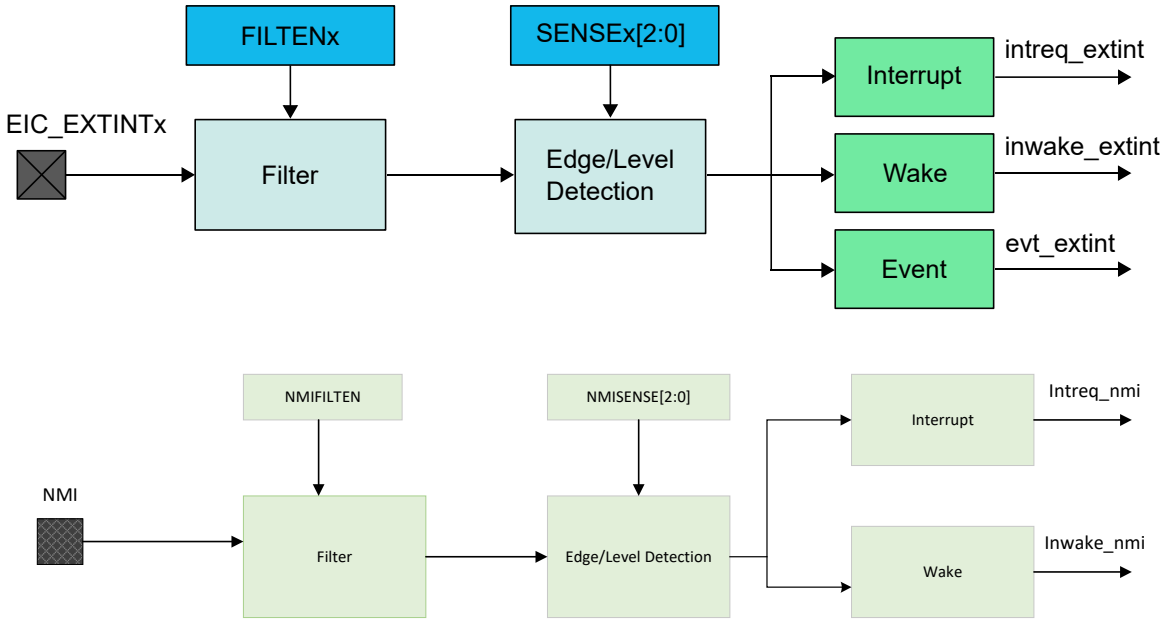
## 28.2. Features

The following are key features of the module:

- Up to 16 external interrupt pins (EIC_EXTINTx, x = 0-(n-1); Where n=16)
- 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 interrupt, EIC_EXTINTx, pins
- Event generation trigger events from EIC_EXTINTx pins

## 28.3. Block Diagram

**Figure 28-1.** EIC Block Diagram





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

## 28.4. Signal Description

| Signal Name | Type | Description |
|---|---|---|
| EIC_EXTINTx[(n-1)..0]; Where n=16 | Digital Input | External interrupt pin |
| NMI | Digital Input | Non-maskable interrupt pin |

**Note:** One signal may be available on several pins.

## 28.5. Product Dependencies

### 28.5.1. I/O Lines

Using the EIC's I/O lines requires the I/O pins to be configured.

### 28.5.2. 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.

### 28.5.3. Clock

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 the Synchronization section for further details.

### 28.5.4. Interrupts

The EIC has the following interrupt sources: External interrupt pins (EIC_EXTINTx; x = 0-(n-1); Where n=16).

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. Each interrupt, 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. Refer to the INTFLAG register for additional information on how to clear interrupt flags. The EIC has at least 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.

### 28.5.5. Events

The EIC can generate the following output events:

- External event from pin EIC_EXTINTx; x= 0-(n-1); Where n=16


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

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 EIC_EXTINTx matches the configuration in the CONFIGn register, the corresponding event is generated, if enabled.

### 28.5.6. 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.

### 28.5.7. 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 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.

## 28.6. Functional Description

### 28.6.1. 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 (EIC_EXTINT) can be filtered using majority vote filtering, clocked by GCLK_EIC or by CLK_ULP32K

### 28.6.2. Basic Operation

#### 28.6.2.1. 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 (NMICTRL) register.
3. Enable GCLK_EIC or CLK_ULP32K when one of the following external interrupt, EIC_EXTINTx pins, modes is selected:
   - EIC_EXTINT pin filtering
   - EIC_EXTINT pin synchronous edge detection
   - EIC_EXTINT pin de-bouncing

   GCLK EIC is used when a frequency higher than 32 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 n register (CONFIG).
5. Optionally, enable the Asynchronous mode.
6. Optionally, enable the Debounce mode.
7. Enable the EIC by writing a '1' to CTRLA.ENABLE.

The following bits are enable-protected, meaning that it can only be written when the EIC is disabled (CTRLA.ENABLE=0):


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

- Clock Selection bit in Control A register (CTRLA.CKSEL).

The following registers are enable-protected:

- Event Control register (EVCTRL).
- Configuration n register (CONFIG).
- External Interrupt Asynchronous Mode register (ASYNCH).
- Debouncer Enable register (DEBOUNCEN).
- Debounce Prescaler register (DPRESCALER).

#### 28.6.2.2. Enabling, Disabling, and Resetting

The EIC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The EIC is disabled by writing '0' to CTRLA.ENABLE.

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.

### 28.6.3. 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 External Interrupt Sense Configuration n 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 EIC_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 n register (CONFIG.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.

**Table 28-1.** Majority Vote Filter

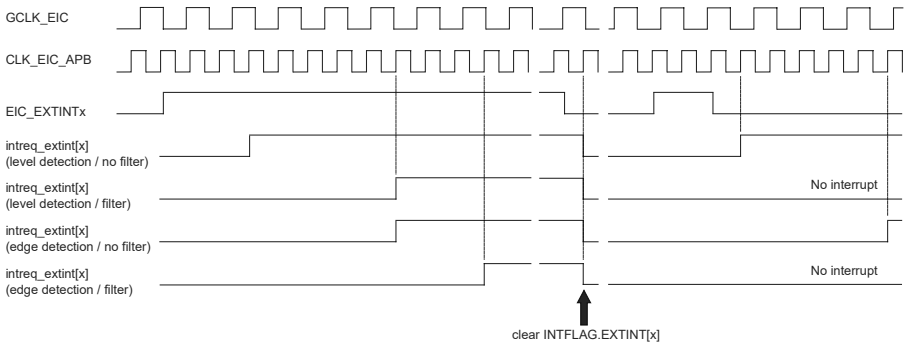
| Samples [0, 1, 2] | Filter Output |
|---|---|
| [0,0,0] | 0 |
| [0,0,1] | 0 |
| [0,1,0] | 0 |
| [0,1,1] | 1 |
| [1,0,0] | 0 |
| [1,0,1] | 1 |
| [1,1,0] | 1 |
| [1,1,1] | 1 |

When an external interrupt is configured for level detection and when filtering is disabled, detection is done asynchronously. Level detection and asynchronous edge detection does not require GCLK_EIC or CLK_ULP32K, but interrupt and events can still be generated.

If filtering or 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.


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

**Figure 28-2.** Interrupt Detection Latency by Modes (Rising Edge)




The detection latency depends on the detection mode.

**Table 28-2.** 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 |

## 28.6.4. Additional Features

### 28.6.4.1. 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 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.

### 28.6.4.2. Asynchronous Edge Detection Mode (No Debouncing)

The EIC_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 (EIC_EXTINTx) 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]) 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 (EIC_EXTINTx) pins set the External Interrupt flag (INTFLAG.EXTINT[x]) directly. In this mode, the EIC clock is not requested.

The asynchronous edge detection mode can be used in Idle and Standby sleep modes.


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

### 28.6.4.3. Interrupt Pin Debouncing

The external interrupt pin (EIC_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 the EIC_EXTINTx pin 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 EIC_EXTINTx is selected, the filter mode for that pin (CONFIGy.FILTENx) cannot be selected.

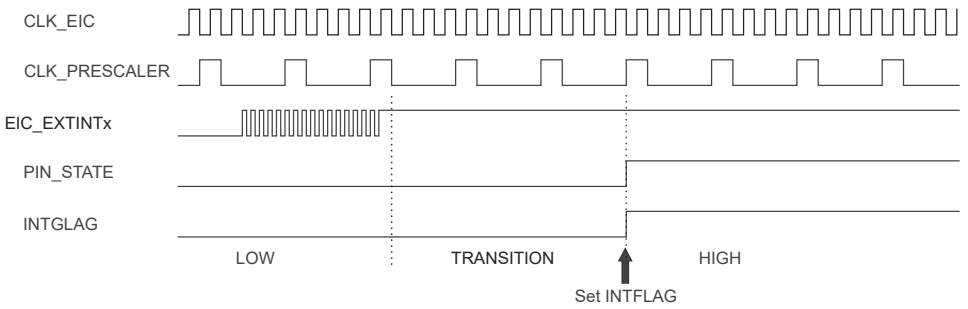
The debouncer manages an internal "valid pin state" that depends on the external interrupt (EIC_EXTINTx) pin transitions, the debouncing mode and the debouncer prescaler frequency. The valid pin state reflects the pin value after debouncing. The external interrupt pin(s) (EIC_EXTINTx) is sampled continuously 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 EIC_EXTINTx pins, 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 PINSTATE register for both synchronous or asynchronous debouncing mode.

Synchronous edge detection – In this mode the external interrupt (EIC_EXTINTx) pin is sampled continuously 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 28-3.** EIC_EXTINTx Pin Synchronous Debouncing (Rising Edge)





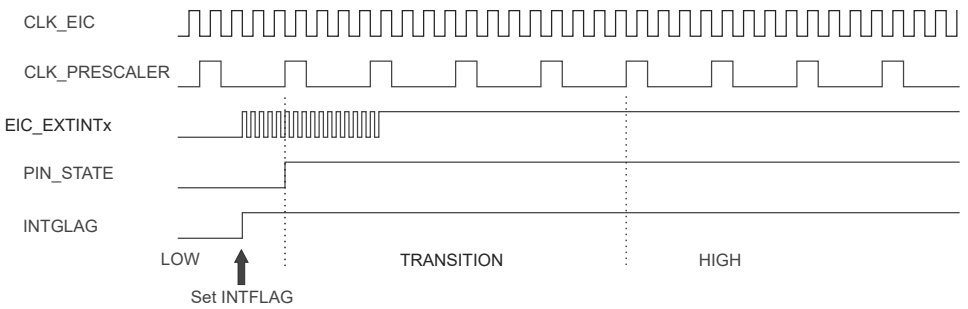
© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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 (EIC_EXTINTx) pins 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 28-4.** EIC_EXTINTx Pin(s) Asynchronous Debouncing (Rising Edge)

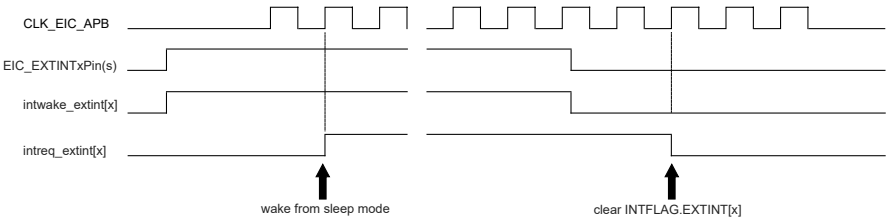



In this mode, the EIC clock is requested. The asynchronous edge detection mode can be used in Idle and Standby sleep modes.

## 28.7. Sleep Mode Operation

In sleep modes, an EIC_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'.

**Figure 28-5.** Wake-up Operation Example (High-Level Detection, No Filter, Interrupt Enable Set)




## 28.8. Secure Access Right

Non-secure write to CTRLA register or DPRESCALER register is prohibited. Non-secure read to CTRLA or DPRESCALER register or SYNCBUSY register will return zero with no error resulting. Non-secure write to a bit of EVCTRL, ASYNCH, DEBOUNCEN, INTENCLR, INTENSET, INTFLAG and CONFIG registers is prohibited if the related bit of NONSEC.EXTINT is zero. Non-secure write to NMICTRL and NMIFLAG registers is prohibited if NONSECNMI.NMI is zero. Bits relating to secure EXTINT read as zero in non-secure mode with no error resulting.


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

## 28.9. Synchronization

Due to the main clock domain and the peripheral clock domains being asynchronous, 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.

## 28.10. Peripheral Dependencies

**Table 28-3.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| EIC | 0x44020000 | NMI : NMI<br>16 : EXTINT0<br>17 : EXTINT1<br>18 : EXTINT2<br>19 : EXTINT3<br>20 : EXTINT4<br>21 : EXTINT5<br>22 : EXTINT6<br>23 : EXTINT7<br>24 : EXTINT8<br>25 : EXTINT9<br>26 : EXTINT10<br>27 : EXTINT11<br>28 : EXTINT12<br>29 : EXTINT13<br>30 : EXTINT14<br>31 : EXTINT15<br>32 : NSCHK | **APB:**<br>CLKMSK[0]<br>MASK13 | PCHCTRL[5] | 13<br>STATUS0<br>PERID13 | | Generators:<br>**EXTINT0:**<br>CHANNEL[21]<br>**EXTINT1:**<br>CHANNEL[22]<br>**EXTINT2:**<br>CHANNEL[23]<br>**EXTINT3:**<br>CHANNEL[24]<br>**EXTINT4:**<br>CHANNEL[25]<br>**EXTINT5:**<br>CHANNEL[26]<br>**EXTINT6:**<br>CHANNEL[27]<br>**EXTINT7:**<br>CHANNEL[28]<br>**EXTINT8:**<br>CHANNEL[29]<br>**EXTINT9:**<br>CHANNEL[30]<br>**EXTINT10:**<br>CHANNEL[31]<br>**EXTINT11:**<br>CHANNEL[32]<br>**EXTINT12:**<br>CHANNEL[33]<br>**EXTINT13:**<br>CHANNEL[34]<br>**EXTINT14:**<br>CHANNEL[35]<br>**EXTINT15:**<br>CHANNEL[36] | PD_CORE_SW |


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

## 28.11. 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 |
| 0x03 | Reserved | | | | | | | | | |
| 0x04 | SYNCBUSY | 7:0 | | | | | | | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | EVCTRL | 7:0 | EXTINTEO7 | EXTINTEO6 | EXTINTEO5 | EXTINTEO4 | EXTINTEO3 | EXTINTEO2 | EXTINTEO1 | EXTINTEO0 |
| | | 15:8 | EXTINTEO15 | EXTINTEO14 | EXTINTEO13 | EXTINTEO12 | EXTINTEO11 | EXTINTEO10 | EXTINTEO9 | EXTINTEO8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C | INTENCLR | 7:0 | EXTINT7 | EXTINT6 | EXTINT5 | EXTINT4 | EXTINT3 | EXTINT2 | EXTINT1 | EXTINT0 |
| | | 15:8 | EXTINT15 | EXTINT14 | EXTINT13 | EXTINT12 | EXTINT11 | EXTINT10 | EXTINT9 | EXTINT8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | INTENSET | 7:0 | EXTINT7 | EXTINT6 | EXTINT5 | EXTINT4 | EXTINT3 | EXTINT2 | EXTINT1 | EXTINT0 |
| | | 15:8 | EXTINT15 | EXTINT14 | EXTINT13 | EXTINT12 | EXTINT11 | EXTINT10 | EXTINT9 | EXTINT8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | INTFLAG | 7:0 | EXTINT7 | EXTINT6 | EXTINT5 | EXTINT4 | EXTINT3 | EXTINT2 | EXTINT1 | EXTINT0 |
| | | 15:8 | EXTINT15 | EXTINT14 | EXTINT13 | EXTINT12 | EXTINT11 | EXTINT10 | EXTINT9 | EXTINT8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | ASYNCH | 7:0 | ASYNCH7 | ASYNCH6 | ASYNCH5 | ASYNCH4 | ASYNCH3 | ASYNCH2 | ASYNCH1 | ASYNCH0 |
| | | 15:8 | ASYNCH15 | ASYNCH14 | ASYNCH13 | ASYNCH12 | ASYNCH11 | ASYNCH10 | ASYNCH9 | ASYNCH8 |
| | | 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 | 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] | | |
| 0x24 ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DEBOUNCEN | 7:0 | DEBOUNCEN7 | DEBOUNCEN6 | DEBOUNCEN5 | DEBOUNCEN4 | DEBOUNCEN3 | DEBOUNCEN2 | DEBOUNCEN1 | DEBOUNCEN0 |
| | | 15:8 | DEBOUNCEN15 | DEBOUNCEN14 | DEBOUNCEN13 | DEBOUNCEN12 | DEBOUNCEN11 | DEBOUNCEN10 | DEBOUNCEN9 | DEBOUNCEN8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | DPRESCALER | 7:0 | STATES | PRESCALERx[2:0] | | | STATES | PRESCALERx[2:0] | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | TICKON |
| | | 31:24 | | | | | | | | |
| 0x38 | PINSTATE | 7:0 | PINSTATE7 | PINSTATE6 | PINSTATE5 | PINSTATE4 | PINSTATE3 | PINSTATE2 | PINSTATE1 | PINSTATE0 |
| | | 15:8 | PINSTATE15 | PINSTATE14 | PINSTATE13 | PINSTATE12 | PINSTATE11 | PINSTATE10 | PINSTATE9 | PINSTATE8 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


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

### 28.11.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronized

**Table 28-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | CKSEL | | | ENABLE | SWRST |
| Access | | | | R/W | | | R/W | 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 kHz is required for filtering) or by CLK_ULP32K (when power consumption is the priority).
This bit is not Write-Synchronized.

| Value | Name | Description |
|---|---|---|
| 0 | CLK_GCLK | The EIC is clocked by GCLK |
| 1 | CLK_ULP32K | The EIC is clocked by 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.


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

### 28.11.2. Non-Maskable Interrupt Control

**Name:** NMICTRL
**Offset:** 0x01
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 28-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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** Asynchronous Edge Detection Mode
The NMI edge detection can be operated synchronously or asynchronously to the EIC clock.

| Value | Name | Description |
|---|---|---|
| 0 | SYNC | Edge detection is clock synchronously operated |
| 1 | ASYNC | Edge detection is clock 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.

### 28.11.3. Non-Maskable Interrupt Flag Status and Clear

**Name:** NMIFLAG
**Offset:** 0x02
**Reset:** 0x0000
**Property:** -

**Table 28-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | NMI |
| Access | | | | | | | | R/W |
| 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.

### 28.11.4. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** -

**Table 28-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 28.11.5. Event Control

**Name:** EVCTRL
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 28-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINTEO15 | EXTINTEO14 | EXTINTEO13 | EXTINTEO12 | EXTINTEO11 | EXTINTEO10 | EXTINTEO9 | EXTINTEO8 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINTEO7 | EXTINTEO6 | EXTINTEO5 | EXTINTEO4 | EXTINTEO3 | EXTINTEO2 | EXTINTEO1 | EXTINTEO0 |
| Access | R/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 – EXTINTEO** 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.

### 28.11.6. 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).

**Table 28-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT15 | EXTINT14 | EXTINT13 | EXTINT12 | EXTINT11 | EXTINT10 | EXTINT9 | EXTINT8 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT7 | EXTINT6 | EXTINT5 | EXTINT4 | EXTINT3 | EXTINT2 | EXTINT1 | EXTINT0 |
| Access | R/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 – EXTINT** External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EIC_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 EIC_EXTINTx pin(s).

| Value | Description |
|---|---|
| 0 | The external interrupt EIC_EXTINTx is disabled. |
| 1 | The external interrupt EIC_EXTINTx is enabled. |


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

### 28.11.7. 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.

**Table 28-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT15 | EXTINT14 | EXTINT13 | EXTINT12 | EXTINT11 | EXTINT10 | EXTINT9 | EXTINT8 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT7 | EXTINT6 | EXTINT5 | EXTINT4 | EXTINT3 | EXTINT2 | EXTINT1 | EXTINT0 |
| Access | R/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 – EXTINT** External Interrupt Enable
The bit x of EXTINT enables the interrupt associated with the EIC_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 EIC_EXTINTx pin(s).

| Value | Description |
|---|---|
| 0 | The external interrupt EIC_EXTINTx is disabled. |
| 1 | The external interrupt EIC_EXTINTx is enabled. |


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

### 28.11.8. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** -

**Note:**

1. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 28-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT15 | EXTINT14 | EXTINT13 | EXTINT12 | EXTINT11 | EXTINT10 | EXTINT9 | EXTINT8 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | EXTINT7 | EXTINT6 | EXTINT5 | EXTINT4 | EXTINT3 | EXTINT2 | EXTINT1 | EXTINT0 |
| Access | R/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 – EXTINT** External Interrupt
The flag bit x is cleared by writing a '1' to it.
This flag is set when the EIC_EXTINTx pin matches the external interrupt sense configuration and will generate an interrupt request if INTENCLR.EXTINT[x] or INTENSET.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.

### 28.11.9. External Interrupt Asynchronous Mode

**Name:** ASYNCH
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected,

**Table 28-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | ASYNCH15 | ASYNCH14 | ASYNCH13 | ASYNCH12 | ASYNCH11 | ASYNCH10 | ASYNCH9 | ASYNCH8 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | ASYNCH7 | ASYNCH6 | ASYNCH5 | ASYNCH4 | ASYNCH3 | ASYNCH2 | ASYNCH1 | ASYNCH0 |
| Access | R/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 – ASYNCH** 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.

### 28.11.10. External Interrupt Sense Configuration

**Name:** CONFIGn
**Offset:** 0x1C + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** RW

**Table 28-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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]

| Value | Description |
|---|---|
| 0 | Filter is disabled for EXTINT[n*8+x] input. |
| 1 | Filter is enabled for EXTINT[n*8+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[n*8+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.

### 28.11.11. Debouncer Enable

**Name:** DEBOUNCEN
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 28-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | DEBOUNCEN 14 | DEBOUNCEN 13 | DEBOUNCEN 12 | DEBOUNCEN 11 | DEBOUNCEN 10 | DEBOUNCEN 9 | DEBOUNCEN 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 |
|---|---|---|---|---|---|---|---|---|
| | DEBOUNCEN 7 | DEBOUNCEN 6 | DEBOUNCEN 5 | DEBOUNCEN 4 | DEBOUNCEN 3 | DEBOUNCEN 2 | DEBOUNCEN 1 | DEBOUNCEN 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 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – DEBOUNCEN** Debouncer Enable
The bit x of DEBOUNCEN set the Debounce mode for the interrupt associated with the EIC_EXTINTx pin(s).

| Value | Description |
|---|---|
| 0 | The EIC_EXTINTx edge input is not debounced. |
| 1 | The EIC_EXTINTx edge input is debounced. |


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

## 28.11.12. Debouncer Prescaler

**Name:** DPRESCALER
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 28-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| 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 |
|---|---|---|---|---|---|---|---|---|
| | STATES | PRESCALERx[2:0] | | | STATES | PRESCALERx[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 16 – TICKON** Pin Sampler frequency selection
This bit selects the clock used for the sampling of bounce during transition detection.

| Value | Name | Description |
|---|---|---|
| 0 | CLK_GCLK_EIC | Bounce sampler is using GCLK_EIC |
| 1 | CLK_LFREQ | Bounce sampler is using Low Frequency Clock |

**Bits 3,7 – STATES** Debouncer number of states

| Value | Name | Description |
|---|---|---|
| 0 | LFREQ3 | 3 low frequency samples |
| 1 | LFREQ7 | 7 low frequency samples |

**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 |
|---|---|---|
| 0 | DIV2 | EIC clock divided by 2 |
| 1 | DIV4 | EIC clock divided by 4 |
| 2 | DIV8 | EIC clock divided by 8 |
| 3 | DIV16 | EIC clock divided by 16 |
| 4 | DIV32 | EIC clock divided by 32 |


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

| Value | Name | Description |
|---|---|---|
| 5 | DIV64 | EIC clock divided by 64 |
| 6 | DIV128 | EIC clock divided by 128 |
| 7 | DIV256 | EIC clock divided by 256 |


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

### 28.11.13. Pin State

**Name:** PINSTATE
**Offset:** 0x38
**Reset:** 0x00000000
**Property:** -

**Table 28-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PINSTATE15 | PINSTATE14 | PINSTATE13 | PINSTATE12 | PINSTATE11 | PINSTATE10 | PINSTATE9 | PINSTATE8 |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | PINSTATE7 | PINSTATE6 | PINSTATE5 | PINSTATE4 | PINSTATE3 | PINSTATE2 | PINSTATE1 | PINSTATE0 |
| 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 – PINSTATE** Pin State
These bits return the valid pin state of the debounced external interrupt pin, EIC_EXTINTx.


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

# 29. Non-Volatile Memory Controller (NVMCTRL)

The NVM Controller is built from two separate modules, described below, the Flash Controller Write (FCW) and Flash Controller Read (FCR).

## 29.1. Flash Controller, Write

### 29.1.1. Features

The following are key features of the Flash Controller, write module:

- Support Flash Partitions
  - Program Flash Memory (PFM)
  - Boot Flash Memory (BFM)
  - Configuration Flash Memory (CFM)
- Mission Mode Write/Erase Features:
  - PFM Erase
  - Page Erase
  - Single Write (32-bit) + simple parity
  - Row Write + SECDED ECC
    - Built-in DMA of data from DRAM
  - All write operations support pre-program to increase endurance and retention
- PFM Address Swap
- Debug Erase features are implemented by ROM Code:
  - Identical to Mission Mode – PLUS:
  - Chip Erase: PFM and unprotected NVR pages - available to the Boot only
  - NVR Test pages cannot be erased
- Debug Programming Features
  - Identical to Mission Mode
- Program Flash Memory (PFM) Write Protect, with 4 protection regions
  - Region defined by Base and Size
  - Lockable
- Boot Flash Memory (BFM) Write Protect
  - Individual (4 KB) Page write protect regions
  - Lockable
- Flash-based user OTP pages
  - Not erased by Chip Erase or Page Erase
  - 4 Lockable Write Protect regions per Page
  - Supports storage of user calibration values

### 29.1.2. Definitions

The Flash Controller, Write (FCW) supports programming operations (writes/erases) for on-chip Flash memory. Most memories have extra bits which the FCW uses to implement Error Correction Code (ECC) for Single Error Correction (SEC) and Double Error detection (DED). For the purpose of the Flash Controller a “Word” is defined as 32-bits. A Write Word (WW) is the native write width of the Flash and is either a Single Word or a Double Word, DWord. “Flash Words” as described in this

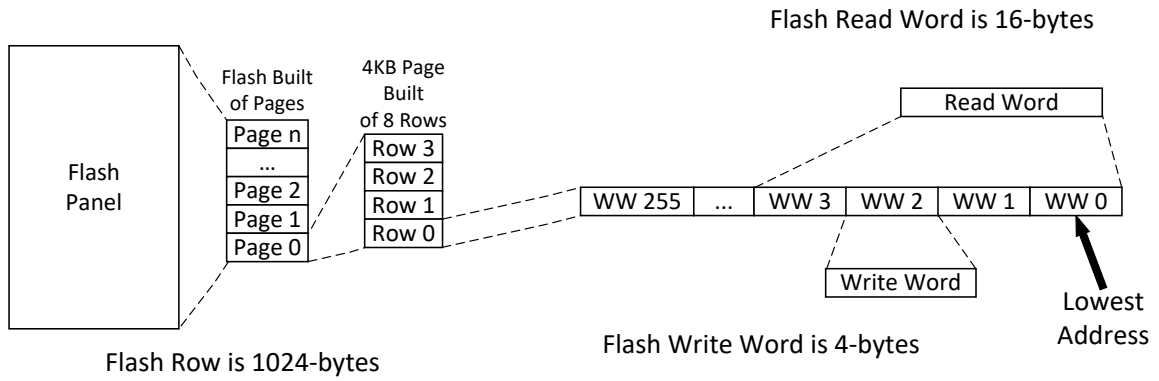

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

section are defined as the native read width of the flash. Flash Words are either Quad Words or QWords .

Flash panels are divided into two logical Flash partitions:

- Program Flash Memory (PFM)
- Configuration Flash Memory (CFM)
- Boot Flash Memory (BFM)

**Figure 29-1.** Flash Module Construction




### 29.1.3. Flash Architecture

Each Flash partition is built up of several pages, also known as sectors. The controller works with a panel made from 4 KB pages with each page containing 4 rows of Flash data. A row is the largest selectable region for contiguous programming of Write Words.

A 128-bit Flash has a 4 row page and a 32-bit write word:

- Row = 256 Write Words

A page of Flash is the smallest unit of memory that can be erased in a single operation. A panel's Program Flash Memory (PFM) space can also be erased in a single operation. All other erases use multiple operations.

A Write Word is the only unit of memory that can be programmed at a time. All other programming operations are made up of several contiguous Write Word Program operations.


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

The FCW supports the following:

- **Row Write:** Write Word by Write Word programming until the whole Row is written. Data is read by the FCW from System SRAM.
- **Quad Write:** The FCW performs 4 writes of data from holding registers.
- **Single Write:** The FCW performs one write of data from holding registers.

### 29.1.4. Boot Support

The NVM Controller supports the following boot methods:

- Single contiguous Boot region using the total BFM size

Additionally, a Secure Boot page can be setup within each Boot region.

### 29.1.5. Program Flash Memory (PFM)

PFM is the largest section of Flash memory. It is where the main application code resides. When a series of parts have the same feature but different Flash size, the PFM is the only partition that changes size.

### 29.1.6. Boot Flash Memory (BFM)

There are two types of Flash memories, Large Flash and Small Flash. The difference between the two is not based on the amount of PFM but the number of pages allocated to Boot Flash Memory. Large Flash memories are meant to support sophisticated boot loaders and hence have 16 pages in each panel's BFM. Small Flash memories are meant to support simple boot loaders and have 4 pages in each panel's BFM.

PIC32C devices have a Boot ROM to control the loading of configuration and to provide a root of trust for secure boot. When the Boot ROM is finished it sets up the CPU to start execution from the base of BFM.

### 29.1.7. Configuration Flash Memory (CFM)

CFM contains eight pages dedicated to configuration. These pages have hardware imposed restrictions on their usage. They are not for code and may not be accessible after the device is protected.

The user OTP pages implements a Flash-based One Time Programmable region. It is always erase protected by hardware such that its content survives a chip erase or page erase operation. It is intended to store calibration values for external devices but is fully user defined.

The Boot Configuration pages contain boot data used by the Boot ROM to configure and boot the device. These pages can be read and write, erase protected, and/or locked.

The ROM Configuration page contains ROM options also for pre-boot usage. This page holds data and configuration that should only be erased when erasing all other pages (PFM, BFM and user accessible CFM pages.)

CalOTP contains device calibration data including the unique ID of the device and manufacturing flow data, respectively. This page cannot be written or erased.

The figures below illustrate the conventional names of the pages in CFM. The register bit descriptions reference these names for feature control.

### 29.1.8. Flash Retention and Endurance

Flash panels have retention and endurance values native to the process technology in which they are designed.

The following mechanisms exist to increase the endurance while maintaining retention, ECC, and Pre-Program:


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

- ECC masks native error rates by detection and correction. Since ECC can correct one error per Flash Word, it can double the endurance of a panel.
- Pre-Program reduces stress when programming bit cells thereby doubling endurance. However, Pre-Program adds about 20 percent to the Flash programming time.

Using both ECC and Pre-Program can yield a significant endurance improvement. For additional information, refer to the *Electrical Specifications* section.

### 29.1.9. ECC Support

The FCW always writes ECC values for Double Word, Quad Word, and Row Write operations. The FCW always writes Parity for Single Write operations. The Flash ECC settings (in FCR ECCCTRL) determine if Single Write operations are allowed. Single Write operations are not allowed when Flash ECC settings are set to ECC Writes with ECC Reads. For all Dynamic Write ECC selections, the FCW allows Single Write operations.

**Note:** The width of the Flash determines if Single Word, Double Word, or Quad Word programming is supported. Refer to the Control Operation Register (CTRLOP) for Flash operations supported for each Flash bit width.

### 29.1.10. Debug Access Level (DAL)

The DAL is fully supported by the Boot ROM and the FCW is DAL agnostic.

### 29.1.11. SFR Write Protect Features

Several mechanisms are in place within the device to prevent accidental writes to program Flash. These mechanisms include:

- Module write protection.
- An unlock value (KEY) to allowing writing critical registers.
- Local SFR bits that provide write locks for other SFR bits.
- SFR bits that provide write/erase protection of Flash memory regions.

The following section covers only the first three items.

#### 29.1.11.1. SFR Write Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC) or the WPCTRL. Write protection does not apply to HSM if present, or when accessed by an external debugger.

See the Control Operation (CTRLOP) register.

#### 29.1.11.2. SFR Write Unlock Values

The KEY register provides write protection to critical NVM registers. KEY uses multiple KEY Values to unlock access to classes of registers. Writing the specific KEY Value to the KEY register unlocks access to the defined set of registers. When the KEY Value protects a multitude of registers the unlock remains in effect until SW clears it. When the KEY Value protects a single register, the KEY clears after a successful write.

KEY is always cleared by a write to a KEY protected register that is currently locked. This causes the INTFLAG.KEYERR bit to be set. SW can clear the unlock state by writing any invalid key to it (recommend all 0's). The INTFLAG.KEYERR is not set by this case.

See the Control Operation (CTRLOP) register for a summary of registers and which ones require a KEY Value to unlock them.

#### 29.1.11.3. Local Lock Bits

Some SFR bits have associated Local Lock bits. These bits lock updates of the associated feature until the next reset. The Lock bits are found in SWAP, PWP, LBWP, UBWP, UOWP, and CWP.


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

#### 29.1.11.4. FCW Sequencer User Model

##### 29.1.11.4.1. Sequencer Enable Control

KEY must be written with the WRKEY value before CTRLOP can be written to start a write/erase sequence. Once CTRLOP is written, the KEY register returns to 0's (locks) and all registers in the FCW are write protected (BUSY = 1) until the operation finishes. Enabling write protection for the registers ensures that addresses, data, and configurations remain unchanged during a sequence.

##### 29.1.11.4.2. Start Sequencer

Once the NVM system is configured, the only registers necessary for programming are CTRLOP, KEY, ADDR, and either DATAn or SRCADDR. Page Erase does not need DATAn or SRCADDR, and other erases do not need ADDR.

To perform Write/Erase operations, follow these steps:

1. Setup ADDR and if programming either DATAn (Single/Quad) or SRCADDR (Row Write).
2. Write WRKEY to KEY.
3. Write <Desired NVMOP> to CTRLOP.
4. The FCW generates an interrupt when it clears STATUS.BUSY and sets INTFLAG.DONE.

When CTRLOP is written, the write/erase sequence starts and the CPU is unable to execute from the selected Flash panel for the duration of the sequence.

### 29.1.12. Flash Write Protect Features

Flash write protect features are for safety not security. They protect against inadvertent erases or updates due to execution of errant code.

#### 29.1.12.1. PFM Write Protect (PWP)

The PWP register provides write/erase protection for a region in PFM using PWP*n*.BASE and PWP*n*.SIZE. BASE is the Flash physical address of the start of the region to be protected. BASE is aligned to the page size of the Flash panel. SIZE is the number of pages to be protected. There are 4 regions that can be protected.

Protection can be enabled or disabled using the PWP*n*.PWPEN bit. However, any address range not defined by PWP*n* is not protected. When PWP regions with different protections overlap, the one with the more restrictive wins. Because of this, overlapping regions are not useful.

See the Interrupt Flag Summary Table for the effects of write protect errors on INTFLAG.

##### 29.1.12.1.1. PFM Write Protect Mirror

With dual-panel systems, PWP can be mirrored from the lower PFM to the upper PFM or from the upper PFM to the lower PFM. When PWP*n*.PWPMIR = 1, PWP*n* is applied to both upper and lower PFM regions. The check simply masks the most significant address bit (BASE, SIZE, ADDR) used in the compare.

Individual mirror bits per PWP*n* regions allows the lower PFM to be protected while the upper region is not (or vice versa).

##### 29.1.12.1.2. PWP Example

For example, to protect a 104 KB region starting at 0x0134_7000:

BASE = 0x0134_7000 << 8 = 0x3470_0000

SIZE = 104/4 = 26 = 0x1A = 0x0000_001A

PWPEN = 0x0000_8000

PWPn = BASE | PWPEN | SIZE

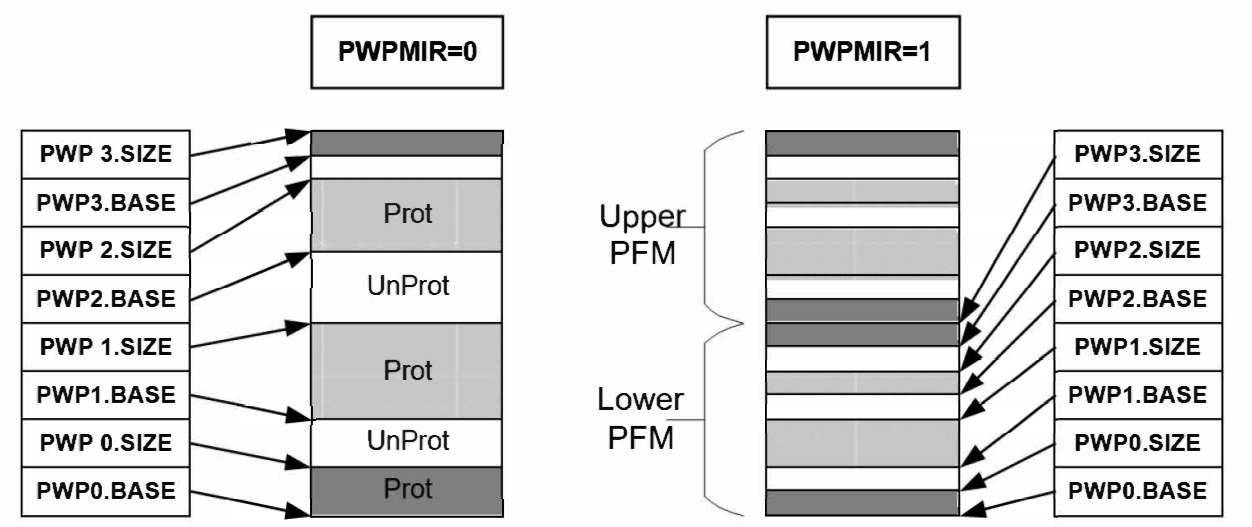
The result for PWPn is 0x3470_801A


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

The following figure shows a graphical representation of Program Write Protect (PWP). Shaded regions are write/erase protected and non-shaded regions are not protected.

- Values for PWPMIR: Mirror PWP bit
- 1 (MIRROR) = PWP settings are Mirrored.
- 0 (UNMIRROR) = PWP settings are NOT Mirrored, also applies to single panel systems.

**Figure 29-2.** PWP Example: No Mirror versus Mirror




#### 29.1.12.1.3. PWPx Lock/Unlock Sequence

PWP*x* registers require a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to the PWP*x* register when locked has no effect.

#### 29.1.12.1.4. Local Lock Bits

The user may optionally select to prevent further writes to a PWP*n* register by also setting the local LOCK bit (in the same register) when writing the PWP*n* values.

When the local lock bit is set (LOCK = 1), subsequent writes to that register (even if the register unlock sequence is followed) have no effect, creating a “write once’ register. Local LOCK bits revert to the unlocked state at reset.

### 29.1.12.2. Boot Write Protect

**Note:** This section uses BWP to refer to both LBWP and UBWP.

Write protection also includes erase protection

The BWP registers are used to write/erase protect individual pages within BFM. The protect bits are grouped into Lower Boot (LBWP*n*) and Upper Boot (UBWP*n*). These are two equal size regions containing the same number of pages. BWP always uses the logical page location which depends on BFSWAP to control protection for the physical page.

At reset, all BWP bits are set to logic ‘1’, write protecting all user accessible BFM pages. Clearing a BFM page bit within its register removes write protect from the corresponding page.

See the Interrupt Flag Summary Table for the effects of write protect errors on INTFLAG.

#### 29.1.12.2.1. BWP Lock/Unlock Sequence

LBWP and UBWP require a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to these registers when locked will have no effect.


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

#### 29.1.12.2.2. Local Lock Bits

The user may optionally select to prevent further writes to a BWP register by also setting the local LOCK bit (in the same register) when writing the BWP value.

When the local lock bit is set (LOCK = 1), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a “write once’ register. The local LOCK bit reverts to the unlocked state at reset.

See Lower (LBWP) and Upper (UBWP) BFM Write Protect Registers for all possible settings regarding the lock and protection bits.

### 29.1.12.3. Configuration Write Protect

The CWP register is used to write/erase protect individual pages within CFM.

At reset, all CWP bits are set to logic ‘1’, write protecting all user accessible CFM pages. Clearing a CFM page bit within its register removes write protect from the corresponding page.

The CWP register contains write/erase protection control for Configuration Flash Memory (CFM).

Some pages are always protected from write/erase commands as noted in the access type for the CWP bit.

#### 29.1.12.3.1. CWP Lock/Unlock Sequence

This register requires a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to this register when locked will have no effect.

#### 29.1.12.3.2. Local Lock Bits

The user may optionally select to prevent further writes to a CWP register bit by also setting the Local Lock bit (in the same register) when writing the CWP value.

When the Local Lock bit is set (LOCK = 1), subsequent writes to that register bit (even if the unlock sequence is followed) have no effect, creating a “write once’ register bit. Local Lock bits revert to the unlocked state at reset.

Some bits cannot be modified but show the protection state of the CFM page.

### 29.1.12.4. User-OTP Write Protect

One page in the Flash Boot region is implemented as One-Time Programmable NVM called UserOTP. It can only be written and cannot be erased (by page or chip erase). It is intended for the storage of user system calibration data that must survive a Flash erase.

UserOTP can be used to store and preserve identification values, such as an Ethernet MAC addresses, OEM/ODM version numbers, board version numbers, and so on. Another use would be for calibration values of circuits external to the chip but constant for a board design in which the chip exists.

The UOWP register is used to write protect 1/4 sized regions of the UserOTP pages. UOWP.UO1WP0 write protect the lowest address quarter page in Panel 1 while UOWP.UO2WP3 write protects the highest addressed quarter page in panel 2.

#### 29.1.12.4.1. UOWP Lock/Unlock Sequence

The UOWP register requires a lock/unlock sequence as described in SFR Write Unlock Values. Any attempt to write to this register when locked has no effect.

#### 29.1.12.4.2. Local Lock Bits

The user may optionally select to prevent further writes to a UOWP regions bits by also setting the Local Lock bits (in the same register) when writing the UOWP value. There are individual Lock bits per ¼ region.

When a local lock bit is set (LOCK = 1), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a “write once’ register. Local LOCK bits revert to the unlocked state at reset.


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

### 29.1.13. Write Sequences

Reads from the panel being programmed stall until the programming sequence is complete. This happens because FCW programming requests have a higher priority than read requests.

**Note:** Performing any programming sequence when power is lost results in an undefined increased wear of the panel. Loss of power is defined as receiving a POR event.

#### 29.1.13.1. Pre-Program

The Flash panels in this device support an option to Flash programming that increase endurance of the panel. This feature is called Pre-Program. When selected, the controller automatically initiates two programming cycles per write. The first is a short cycle and the second is a full cycle. This option adds about 20% to the programming time.

To use this feature, set the CTRLOP.PREPG SFR bit when writing CTRLOP.NVMOP. In all other respects the SFR setup is identical. Pre-Program typically doubles the native endurance and retention of the panel. It is recommend to either always use or never use Pre-Program feature. Mixed use of PREPG yields undefined endurance and retention of the panel.

It is highly recommended to always have CTRLOP.PREPG = 1.

#### 29.1.13.2. Single Write Sequence

The smallest block of data that can be programmed with a single write is one Word. The data to be programmed must be written to DATA0 before the programming sequence is initiated. The FCW programs the DATAn into the Word/DWord location pointed to by ADDR. ADDR is a byte address, therefore the lower 2 or 3 bits are ignored for Word or DWord programming.

A Word/DWord Program sequence typically comprises the following steps:

1. <Desired NVMOP> is Single Write.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.

A Word/DWord within the Flash can be programmed if its associated page write protection is not enabled, see Debug Access Level.

##### 29.1.13.2.1. Single Write Timing

The Word/DWord Program timing is dominated by setup (Tnvs + Tpgs), program time (Tprog), and recovery (Trcv) delays. Using the timing, shown in “Flash Timing Parameters”, total time to program one Word is roughly shown below:

Word/DWord Program time = Tnvs + Tpgs + Tprog + Trcv + Trw

If Pre-Programming, the first step adds roughly:

Word/DWord Program time = Tpreprog + Tprepgh + Tprepgs

**Note:** For write timing, see the *Non-Volatile Memory Controller (NVM) Electrical Specifications* section in the *Electrical Characteristics* chapter.

##### 29.1.13.2.2. Single Write and ECC

Single writes are not allowed because of the Flash ECC setting. The Interrupt Error Flag Summary table defines the interrupt flags that are set.

#### 29.1.13.3. Quad Write Sequence

The Quad Write operation implements a Quad Word programming. The 4, 32-bit instruction/data words to be programmed must be written to DATA*n*, *n* = 0,1,...,3, before the programming sequence is initiated. The Flash instruction word at the location pointed to by ADDR is programmed. The program operation must be aligned to a Flash word address, so the lower 5 bits are ignored for Quad Write.


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

The Quad Write sequence typically comprises the following steps

1. <Desired NVMOP> is Quad Write.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.
5. Unlock the hardware write mutex by setting the LOCK bit to '0' and the OWNER field to '00' simultaneously to the MUTEX register.

A Flash address can be written if its associated page write protection is not enabled, see Debug Accesss Level. The FCW writes whatever is in DATA*n* when commanded.

#### 29.1.13.3.1. Quad Write Timing

Quad Write timing is dominated by setup (Tnvs + Tpgs), program time (Tprog), and recovery (Trcv) delays. Using the specifications shown in *Non-Volatile Memory Controller (NVM) Electrical Specifications*, the total time to program one QWord/QDWord is roughly:

Quad Write Time = Tnvs + Tpgs + 4*Tprog + Trcv + Trw

If Pre-Programming, the first step adds roughly:

Quad Write Time = 4*Tpreprog + Tprepgh + Tprepgs

Quad Write asserts programming voltage throughout the sequence to avoid duplicating setup and recovery delays. Consequently, a Quad Write operation takes less time than 4 Word/DWord Program operations.

#### 29.1.13.3.2. Quad Write and ECC

**Note:** All configuration data held in Flash and read before user boot up must be programmed by software, using Quad Write or it will generate an ECC error when read.

### 29.1.13.4. Row Write Sequence

The largest block of data that can be programmed by a single NVMOP command is a row. As shown in Flash Module Construction, a row is 1024 bytes of data. ADDR is the row aligned address where the Flash address starts programming the data. The controller ignores the sub-row address bits and always starts programming at the beginning of a row.

A Row Write sequence comprises the following steps:

1. Write the entire row of data to be programmed into the system SRAM. The source address must be Word aligned and secure/non-secure zone consistent with the transaction type.
2. <Desired NVMOP> is Row Write.
3. Follow the Start Sequencer from *Start Sequencer.*
4. Wait for NVM Interrupt from *Interrupts.*
5. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See *Errors and Flags* regarding error flags.

A row of Flash can be programmed if its associated page is not write protection. For additional information, see *Debug Access Level.*

The data transferred from RAM is buffered within the FCW. The FCW fetches enough data Words to calculating ECC for each programming operation in the row.

The FCW automates programming the data into the Flash using contiguous Single Write operations. The data is double buffered, such that each programming sequence of four programming operations may execute while the next data are read from the internal system SRAM.


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

**Notes:**

1. The base address held in the SRCADDR register is not changed during the Row Write sequence.
2. The DATAn registers may be affected by the Row Write sequence and cannot be guaranteed to hold prior data.

#### 29.1.13.4.1. Row Write Timing

Row Write timing is dominated by setup (Tnvs + Tpgs), program time (Tprog), and recovery (Trcv) delays.

Row Write timing is dominated by setup (Tnvs + Tpgs), program time (Tprog), and recovery (Trcv) delays. Writes per row, *n*, are based on the number of rows per page and the size of a write:

- BytesPerRow = 512
- BytesPerWrite = 4
- n = BytesPerRow/BytesPerWrite

Using the specifications shown in *Non-Volatile Memory Controller (NVM) Electrical Specifications*, the total time to program one row is roughly:

Row Write Time = Tnvs + Tpgs + n*Tprog + Trcv + Trw

If Pre-Programming, the first step adds roughly:

Row Write Time = n*Tpreprog + Tprepgh + Tprepgs

Row Write asserts programming voltage throughout the sequence to avoid duplicating setup and recovery delays.

#### 29.1.13.4.2. Row Write and ECC

Row Write has the same restrictions as programming operations that support ECC.

### 29.1.14. Erase Sequences

Any access to the panel containing the page or panel being erased stalls reads from all panels until the erase finishes.

#### 29.1.14.1. Page Erase Sequence

A page erase performs an erase of a single page of PFM or BFM, or select pages in CFM (User, Boot), as long as it is not write protected. The page to be erased is selected using ADDR. The lower bits of the address given by ADDR are ignored in page selection.

**Note:**

1. During the page erase, accessing any uncached or unbuffered address within the panel that contains the page, causes that access to stall until the erase finishes. It is the responsibility of user's code to prevent or handle that situation.
2. It is also the responsibility of the programmer's code to ensure that no bus initiators, including the CPU, access the Flash region, the entire PFM of the panel, or the entire Boot of the panel, that is to be (or is being) erased.

##### 29.1.14.1.1. Page Erase

A page erase sequence comprises the following steps:

1. <Desired NVMOP> is Page Erase.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.

A page of Flash can be erased if its associated page write protection is not enabled, see Debug Access Level.


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

#### 29.1.14.1.2. Page Erase Timing

The page erase timing is dominated by setup (Tnvs), erase time (Terase), and recovery (Trcv) delays. Using the timing shown in *Non-Volatile Memory Controller (NVM) Electrical Specifications*, the total time to erase one page is:

Time to Erase Page = $T_{nvs}$ + $T_{erase}$+ $T_{rcv}$+ $T_{rw}$

#### 29.1.14.1.3. Page Erase Retry

The page erase retry is no longer needed to achieve 20K cycle endurance, use Pre-Program for 20K.

### 29.1.14.2. Upper and Lower PFM Erase

The Upper PFM erase NVMOP command erases the PFM in the panel mapped to the Upper PFM address range. The Lower PFM erase NVMOP command erases the PFM in the panel mapped to the Lower PFM address range. These commands leave the BFM and CFM pages intact, and are intended to be used by a field upgradeable device. To guard against accidental erasure, user code can setup PWP for protection. The ability to erase the PFM in the Upper or Lower region allows executing from one panel while erasing (or writing) the other, without the need to complete many Page Erase operations.

A Panel Erase sequence comprises the following steps:

1. <Desired NVMOP> is both Upper and Lower PFM Erase.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.

The PFM in a panel mapped to the Upper or Lower PFM region can be erased if write protection is not enabled for addresses associated with that region.

If in mission mode, the application must not be executing from the erased region. If any initiator reads from the panel containing the region being erased, all Flash accesses are stalled until the erase completes.

### 29.1.14.3. PFM Erase

The PFM Erase NVMOP command erases the entire PFM region. This mode leaves all BFM and CFM pages intact, and is intended to be used by a field upgradeable device. To guard against accidental erasure, setup the PWP register appropriately.

The ability to erase the PFM in the Upper or Lower region allows executing from one panel while erasing (or writing) the other, without the need to complete many Page Erase operations.

The PFM Erase operation is not address based (that is, the FCW directly selects the target Flash panel). The ADDR register is ignored, therefore does not need to be initialized for this NVMOP.

A Panel Erase sequence comprises the following steps:

1. <Desired NVMOP> is Upper or Lower PFM Erase.
2. Follow the Start Sequencer from Start Sequencer.
3. Wait for NVM Interrupt from Interrupts.
4. Check the INTFLAG bits to ensure that the program sequence completed successfully, and then clear all bits in INTFLAG. See Errors and Flags about error flags.

The Flash memory in a panel mapped to the Upper or Lower PFM region can be erased if write protection is not enabled for addresses associated with that region, see Debug Access Level.

If in mission mode, the application must not be executing from the erased region. If any initiator reads from the panel containing the region being erased, all Flash accesses are stalled until the erase completes.


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

#### 29.1.14.4. Program Erase Timing

Program Erase timing is dominated by setup (Tnvs), erase time (Terase), and recovery (Trcv) delays.

Program Erase = 2 * (Tnvs + Terase + Trcv + Trw)

**Note:** For write timing, see the *Non-Volatile Memory Controller (NVM) Electrical Specifications section* in the *Electrical Specifications* chapter.

### 29.1.15. Operation Timing

The FCW supports only self-timed operations using an internal clock source. The timing values for writing and erasing the Flash are predefined. No user involvement is required for timing value selection.

The AHB bus host initiator operates independently of the FSM, transferring data to the FCW data buffer only when requested. Consequently, there are no timing restrictions with respect to AHB and the FSM. However, for Row Write operations, the AHB interface must have sufficient bandwidth to SRAM to keep up and therefore not starve the FCW for data.

### 29.1.16. Interrupts

The Flash module has an interrupt output that is controlled by INTENSET and INTENCLR. The interrupt status is reported through INTFLAG. A read of either INTENSET or INTENCLR returns the enable state for the specific interrupt condition. The Flash interrupt is asserted if a condition is enabled and its associated bit is set in INTFLAG. Interrupt flags are cleared by writing a '1' to its bit position in the INTFLAG register.

It is recommended to always enable DONE as interrupt source in INTENSET. Using DONE masks side effects of WRERR, RSTERR, BUSERR, and FIFOERR being set before the FCW has finished cleaning up after an error.

The NVM Busy Status (STATUS.BUSY) indicates if the FCW is performing an operation (from the CPU). When NVMOP completes, the FSM clears BUSY and sets Done.

The NVM Operation Done Flag (INTFLAG.DONE) indicates that the FCW has completed an NVMOP (for the CPU).

**Note:** The FCW does not clear DONE automatically. Also, DONE = 1 does not prevent NVM operations. Therefore, software must clear DONE before the next NVM operation is started.

### 29.1.17. Errors and Flags

Several non-typical events can occur that affect the operation of the FCW causing write/erase operations to be aborted. Setup errors can occur that prevent the start of an NVMOP command. All these errors are reported through INTFLAG. Some flags are updated before the FSM clears BUSY. However, the FSM always clears BUSY and sets DONE on the same clock edge.

The following table describes the types of errors the FCW detects and what flags they affect.

**Table 29-1.** Interrupt Flag Summary

| INTFLAGs: | Source: | Effect on FCW Operations: |
|---|---|---|
| WRERR, DONE | Any Error Event during NVMOP | Last address programmed may be corrupted, or last erase may not be complete. |
| RSTERR(1) | Reset or Low Voltage Event before FSM starts NVMOP | NVMOP not started(2) |
| RSTERR(1), WRERR | Reset or Low Voltage Event after FSM starts NVMOP | Last address programmed may be corrupted, or last erase may not be complete. Software should verify results. |
| WRERR, SECERR, DONE | Security module or tamper event preempted operation | Last address programmed may be corrupted, or last erase may not be complete. |
| KEYERR | KEY not valid for SFR write | SFR not updated |


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

**Table 29-1.** Interrupt Flag Summary (continued)

| INTFLAGs: | Source: | Effect on FCW Operations: |
|---|---|---|
| CFGERR, DONE | ADDR not in Flash memory space, disallowed CPUDAL setting. | NVMOP is not started (2) |
| CFGERR, OPERR, DONE | NVMOP not allowed because of ECC, i.e. Single Write attempted when ECC Write with ECC Read is selected | NVMOP is not started (2) |
| WRERR, FIFOERR, DONE | Row Write FIFO under run error | NVMOP is aborted(2) Subsequent addresses not programmed |
| BUSERR, FIFOERR(3), DONE | AHB Bus Host error before FIFO filled, possible SRCADDR out of range | NVMOP is not started(2) |
| BUSERR, FIFOERR(3), WRERR, DONE | AHB Bus Host error Possible SRCADDR out of range or access permission violation | NVMOP is aborted(2) Current and subsequent addresses not programmed |
| WPERR, DONE | ADDR is write protected by a WP register (i.e BWP, PWP, etc) | NVMOP is not started(2) |
| OPERR, DONE | NVMOP is not available (e.g. RSVD) | NVMOP is not started(2) |

**Notes:**

1. IDONE will not be set because it is cleared by the reset.
2. NVMOP is generally considered *started* when the panel is enabled.
3. Error code may or may not be set.

The following table summarizes events that prevent the write/erase sequence from happening, but do not set the error flag.

**Table 29-2.** Conditions Without an Error Flag

| Source: | Effect on FCW Operations: |
|---|---|
| Writing KEY with an invalid value. | Locks all registers protected by KEY. |
| Attempted write of an SFR while locked for programming (BUSY = 1) from the same security association. | Action is ignored by FSM and SFRs. Reports a bus error. |
| Attempted read/write of an SFR address that is not implemented. | Action is ignored by FSM and SFRs. Reports a bus error. |

### 29.1.17.1. Write Error (INTFLAG.WRERR)

The FCW Finite State Machine (FSM) set the WRERR flag only if the FSM aborts a Write/Erase operation (that is, an NVMOP command). The FSM sets the flag on the detection of an error which is before it clears BUSY and sets DONE. WRERR is only set if there is possible corruption of the Flash due to the NVMOP abort.

### 29.1.17.2. Reset and Low Voltage (BOR) Error (INTFLAG.RSTERR)

If a reset or BOR event occurs while an NVMOP command is underway, the FCW sets flags according to the Interrupt Flag Summary table and aborts row programming (on a quad write boundary) and all erases. A system reset prevents an interrupt from occurring. However, relevant INTFLAG bits are reset on POR and maintain state through a system reset and a BOR. Therefore, software can check the prior state of the FCW. It can also check the device's RCAUSE register to see if the error is from a system reset or from a BOR. If system shows a POR occurred, software has no guaranteed way of knowing if the system had been operational and a write/erase operation was underway at the time the POR occurred.

### 29.1.17.3. Power-on Reset

A Power-on Reset (POR) causes immediate termination of any FCW operation. A POR is a full and hard reset of all states of the FCW and therefore no error flags originating from a prior operation survive the reset.


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

Several SFR bits get reset by POR so they survive a BOR or user resets (WDT Reset, RESET pin, and so on). Doing so allows many NVM operations to complete and the ones that don't can report errors. Also, they prevent values like address from changing in the middle of an operation.

## 29.2. Flash Controller, Read

### 29.2.1. Overview

The Flash Read Controller provides up to 1 AHB system read targets for the Flash, an ECC calculation module, automatic CRC scanning of Flash, and fault injection. It also provides arbitration between the FCW and the system, ready timers for startup from POR/sleep, and multiple panel address decoding.

The FCR provides a user interfaces to Flash read control features. The FCR contains, Flash Wait state control, ECC fault reporting, ECC fault injection, and Flash CRC. The Fault and CRC features that can be used to meet certain functional safety requirements for invariant memories.

For additional information, refer to the Definitions and Flash Architecture.

### 29.2.2. Features

The following are key features of the FCR module:

- Supports Flash partitions
  - Program Flash Memory (PFM)
  - Configuration Flash Memory (CFM)
- Flash-based configuration retention support:
  - Read and update external module calibration and configuration data
  - Read and update Flash IP calibration data
  - Read and update Page Redundancy Records (locally)
- Flash Memory Read Protection for pages in CFM
  - ROM CFG and Boot CFGx
- Advanced High-performance Bus (AHB) reads of Flash memory
  - Up to 1 AHB read ports
  - One read buffer per port
- CRC logic for in-variant memory tests
- Fault logic for error injection and testing

### 29.2.3. Power Management

The Flash system supports multiple power modes for reducing or managing power. These include auto standby, Flash power-down, and controlling power during sleep.

#### 29.2.3.1. Flash Auto Standby

The Flash panel enters a Standby mode automatically after a read or write operation. Each Flash panel enters its Auto-Standby mode independently of the other. After a read request completes as defined by the read access time ($T_{ACC}$), the panel enters Auto-Standby mode. For writes/erases this after the recovery time. When a panel is in this mode read and write requests incur no access time penalty.

#### 29.2.3.2. Flash Hibernate

The Flash panel supports Flash Hibernate mode, which cuts power to most of the panel's circuits. This substantially reduces the power consumption of Flash systems. Flash Hibernate affects both panels.

Flash Hibernate mode incurs an entry and exit delay, $T_{DPDS}$ and $_{TDPDH}$, respectively. When in Flash Hibernate mode all Flash operations are suspended; however, the Flash control system


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

automatically wakes up the panels on a request (read or write). The panels are not automatically put back into Hibernate after a wakeup event. If the interrupt causes a wakeup to sleepwalking the panels return the mode selected by CTRLB.SLP.

See *Power Mode in Standby Sleep* for manual entry and exit of Flash Hibernate.

#### 29.2.3.3. Sleep Interaction

The Flash system holds off entry into sleep modes lower than Idle Sleep to cleanly stop Flash operations. The following operations delay entry into sleep:

- In-progress read from any source (system, CRC, and so on)
- Flash mode change (that is, High Temp, wake from Flash Hibernate)
- Requested program or erase operation, and so on

**Note:** Row programming and erase operations can significantly delay entry into a sleep mode.

Entry into Idle Sleep mode does not cause a Flash mode change. Clocks in the Flash system stop automatically when they are not needed for operation.

Entry into Standby Sleep mode can cause a Flash mode change. The Flash can be setup to enter Auto-Standby or Flash Hibernate. If Flash Hibernate is selected there is the option to wakeup to Auto-Standby or remain in Flash Hibernate until the first access. Waking up to Auto-Standby provides the fastest initial access to the Flash system.

See *Power Mode in Standby Sleep* for power mode control of the Flash on entry and exit of system Standby Sleep.

### 29.2.4. Flash Read Control

#### 29.2.4.1. Read Control Overview

The FCR provides read access to the Flash system for the CPU and peripheral hosts. It implements 1 AHB client. This interface runs at the same frequency as the system bus and uses the same Flash access timing control.

AHB interface 1 supports a prefetch buffer and cache (PCACHE). The size of the cache is 128 by 4 lines. The PCACHE is unified and treats instructions and data the same. The predictive prefetch unit hides the access time of the Flash memory by fetching the instructions from the next sequential address while the prior instructions are being fetched and executed.

Accesses that hit the cache or prefetch buffer are returned with zero wait states. Misses access the Flash and therefore act according to *FWS[3:0] - Flash Access Time Wait States (Defined in terms of AHB Clocks)*.

#### 29.2.4.2. CTRLA SFR Description

The CTRLA fields can be changed at anytime, provided the FCR is not write protected. New wait state values are used on the next Flash read. RDBUFWS changes are used on the next update to the associated read buffer. A new arbitration value is used at the next available arbitration cycle on a per panel basis.

The register values for CTRLA are given below.

**PRIV - Privileged Access Only**

The PRIV field determines if the FCR module accepts all accesses or only privileged accesses. It is set to privileged (1) by default.

**FWS[3:0] - Flash Access Time Wait States (Defined in terms of AHB Clocks)**

The FWS[3:0] field controls the number of extra clock cycles it takes to access the Flash memory and provide data back (to the bus or internal requester). The number of wait states needed is typically Taws/System Clock Period (or AHB Clock Period) rounded up to the nearest integer. If ECC is enabled an extra wait state may be needed. For additional information on the frequency range for each wait state selection, see *Flash Controller Electrical Characteristics*.


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

**ADRWS - Address Wait State Enable**

The ADRWS field controls the number of extra clock cycles for address setup to the Flash memory. The Address Wait state occurs before requesting arbitration, therefore it is in between the bus and the request to Flash. In most systems, the path starts at the initiator of the request.

When ADRWS = 0, no extra cycle clocks are added to address setup. When ADRWS = 1, one extra clock is added to address setup. ADRWS only affects AHB interfaces. It does not affect internal read requests, such as from CRC logic if present.

**AUTOWS - Automatic Wait State Enable**

The AUTOWS field controls the use of Automatic Wait state for the Flash access time (Taws). This is useful when varying clock frequencies.

When AUTOWS=1, the total Flash wait states are ADRWS+Taws. When AUTOWS = 0 the total Flash wait states are ADRWS+FWS. Non-CPU ports may incur an extra wait state. The bus might add pipeline delays to either the address phase or the data phase.

**RDBUFWS[3:0] - Read Buffer Wait States**

The RDBUFWS field controls the determinism of accesses from the read buffer. Each bit corresponds to an AHB Interface 'n'. When RDBUFWS[n] = 0, the associated AHB interface returns data from the read buffer in zero wait states. When RDBUFWS[n] = 1, the associated AHB interface returns data with wait states to match the Flash access of ADRWS+FWS such that the initiator see a constant access time.

The setting of RDBUFWS[n] is ignored when AUTOWS = 1. For this case, all data is returned from the read buffer with zero wait states.

For devices with a Flash cache, RDBUFWS[n] applies to its interface only if the cache is disabled. Accesses to the flash from debug mode are completed through the read buffer.

### 29.2.4.3. CTRLB SFR Description

The CTRLB fields can be changed at anytime, provided the FCR is not write protected. Updates that affect the cache happen on read access boundaries.

The Register values for CTRLB are discussed below.

**PRM - Power Reduction Mode**

The PRM field allows the user to manually control the Flash power mode. Writing CTRLB.PRM to a 1 toggles the power reduction mode. The current mode is reflected in STATUS.PRM. When in Flash Hibernate the panel wakes up on access but does not return to Flash Hibernate. Note that when awake the Flash reenters Auto Standby if not accessed for one read cycle time.

To change the Flash Power Reduction Mode:

1. Read current PRM value. curr_prm = STATUS.PRM.
2. Write 0x01 to BYTE 0 of CTRLB (toggle Flash Power mode).
3. Wait until STATUS.PRM toggles. while (curr_prm == STATUS.PRM) {}.

No flags get set nor are interrupts generated from a power reduction mode change. The power status of the Flash is reflected in STATUS.PRM.

**Note:** Changing the Power Reduction Mode incurs a delay of TDPDH to take effect.

**SLP[1:0] - Power Mode in Standby Sleep**

The SLP field controls the Power Reduction Mode during System Standby, Sleep. Waking the Flash from its power down mode takes at least $T_{DPDH}$. Upon system wakeup, the current Power Reduction Mode is reflected in STATUS.PRM. See the SLP SFR field for more details.


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

## 29.2.5. Cyclic Redundancy Check (CRC) Support

### 29.2.5.1. CRC Overview

The flash system supports automated CRC of any contiguous flash region. It supports both 16-bit and 32-bit CRC with a programmable polynomial, initial value, final XOR value, and checksum compare value. The CRC can run on-command or continuously with auto repeat.

All control registers must be setup prior to enabling CRC operation. Once setup, writing CRCEN = 1 loads the Initial Value into the Accumulator, the Message Length into they Message Length counter, and the Period into the Period counter. After the Period counter decrements to zero the logic reads the data at the Message Address. For each byte shifted through the LFSR, it decrements the byte counter.

Bytes shift into the LFSR lowest addressed to highest addressed. The Reflected Bit Order field, REFIN, determines if it is MSbit first or LSbit first. During the shifting the Accumulator bits toggle showing the result. Once the state machine completely shifts in the last byte from the current flash read, it reloads the Period counter and starts counting.

The CRC logic continues reading, shifting, and waiting until the Message Length counter reaches zero. The final step of the CRC operation XORs the Accumulator with the Final XOR value before comparison to the Check Sum value. Reflected Bit Order field, REFOUT, determine if the Accumulator is reversed or not before the Final XOR.
**Note:** The Accumulator is not updated with the Final XOR value. If the comparison fails logic sets the CRCERR flag and stops with DONE flag also set.

If no error occurs and AUTOR is not set, the CRC logic sets CRCDONE and CRCERR remains cleared. If AUTOR is set, then the CRC logic reloads all initial settings are restarts the CRC calculation without setting CRCDONE.

### 29.2.5.2. Manual Usage Model

When the Auto Repeat bit selects manual mode, the last step of the CRC operation always sets the CRCDONE bit. The state of CRCERR reflects the validity of the message (flash memory region).

If checking the same region (i.e. message) again, simply clear CRCDONE and CRCERR before setting CRCEN again. The IV loads into the ACC and the operation begins again.

### 29.2.5.3. Auto Repeat Usage Model

When the Auto Repeat bit selects repeat mode, the CRC operation only exits at the end of a loop if there is a comparison error. The error causes the logic to set both CRCDONE and CRCERR. To restart the CRC, simply clear CRCDONE and CRCERR before setting CRCEN again.

If there is no comparison error the logic restarts the CRC operation by loading the IV into ACC, the MLEN into the length counter, and Period into the Period counter. CRCDONE, CRCERR, and CRCEN remain unchanged. As mentioned above, each read waits the Period count including the first read of a loop. The CRC operation continues looping until it detects an error or is stopped manually.

### 29.2.5.4. CRC Interrupt Event

The CRCDONE bit reflects the state of the interrupt. When it is ONE the interrupt is asserted. Clearing CRCDONE de-asserts the interrupt. The system level interrupt controller determines if the CPU see the interrupt.

### 29.2.5.5. Performance

To process each byte of data (8-bits), the calculation takes one peripheral clock. However it reads a multi-byte flash word and operates on all those bytes each read. The Flash Word in Bytes, FWB, is either 16 or 32 bytes depending on the flash system. This yields either 16 or 32 clocks to calculate a full FWB.

Since the CRC is reading the Flash it must also use the system Flash Wait States, FWS. The time to get the flash data in to the shift register is FWS+1. Also, if the system continues to access flash, the CRC logic is lowest priority and must wait until there is an idle cycle.


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

The logic periodically reads the flash to supply data to the calculation. The PERIOD is programmable from 0 to 4095 extra Period Clock (PerCLK) counts from the end of the calculation. The PerCLK is an 8MHz fixed frequency clock. Though selecting a period of zero clocks won't completely cut off the system flash bandwidth, it could be severely limiting. System bandwidth can be less than half under such conditions. Low period selections are useful during light loads or scheduled checks. When running with auto repeat, an appropriate period value is one that finishes the calculation before to the maximum user alloted time.

Lastly, the CRC needs to know the Message Length, MLEN, in bytes over which to perform the calculation. Note that neither the starting address of the message nor the message length need to be aligned to FWB.

#### 29.2.5.5.1. CRC Performance Equation

First, calculate the following values:

$T_{MCALC}$ = CRC message calculation time

$P_{APB}$ = 1/(Peripheral Bus speed) = Peripheral Bus Period

$P_{AHB}$ = 1/(AHB Clock speed) = AHB Period

FWB = Flash Word in Bytes = 32

$T_{STEP}$ = FWB*$P_{APB}$ = CRC step time

$T_{PER}$ = $P_{PERCLK}$*PERIOD + 2*$P_{BUS}$ * PERIOD = The Period between CRC flash accesses

For PERIOD > 0, 2*$P_{BUS}$ * PERIOD = 2*$P_{BUS}$

For PERIOD = 0, 2*$P_{BUS}$ * PERIOD = 0

$T_{ACC}$ = (FWS+1)*$P_{BUS}$ = Flash Access Time

$P_{BUS}$ = Bus Clock Period (note: AHB or APB clock as they are the same)

$P_{PERCLK}$ = Period of Period Clock = 1/8 MHz = 125ns

Second, solve the following for the Message Calculation Time:

$T_{MCALC}$ = ($T_{STEP}$ + 2*$T_{ACC}$ + $T_{PER}$)*MLEN/FWB

2*TACC includes the delay waiting for a prior read to finish, and depends on flash utilization. MLEN/FWB is the number of flash reads needed by the CRC engine. If data is not aligned to FWB it may take 1 more read.

If $T_{MCALC}$ is known or has been calculated, we can solve for the PERIOD:

PERIOD = [(FWB/MLEN)*$T_{MCALC}$ - ($T_{STEP}$ + 2*$T_{ACC}$ + 2*$P_{BUS}$)]/$P_{PERCLK}$

(note: $P_{BUS}$ = AHB or APB clock as they are the same)

#### 29.2.5.5.2. CRC Performance Example

Assume the application must calculate the CRC (Period) on an 80Kbyte message every 800ms. Further, assume that both the AHB and APB are running at 100 MHz and that Flash needs four wait states.

First, establish the following:

FWB = 32Bytes

MLEN = 80Kbytes = 80Kb*1024Bytes/KB = 81920Bytes

$T_{MCALC}$ = 800ms = 800,000,000ns

$P_{APB}$ = 1/100MHz = 10ns

$P_{AHB}$ = 1/100MHz = 10ns


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

$P_{PERCLK}$ = 125ns

FWS = 4

MLEN/FWB = 81920Bytes/32Bytes = 2560

FWB/MLEN = 1/2560

$T_{STEP} = FWB*P_{APB} = 32*10ns = 320ns$

$T_{ACC} = (FWS+1)*P_{BUS} = (4+1)*10ns = 50ns$

Second, solve for the value of PERIOD:

$PERIOD = [(FWB/MLEN)*T_{MCALC} - (T_{STEP} + 2*T_{ACC} + 2*P_{BUS})]/P_{PERCLK}$

$= [(1/2560)*800,000,000ns - (320ns + 100ns + 20ns)]/125ns$

$= ((1/2560)*8e8ns - 440ns)/125ns = (312,500 - 440)/125 = 2496.48$

= 2496 (rounding down)

#### 29.2.5.6. Effects of ECC on CRC

CRC reads of the Flash obey the selected ECC mode of the device. If ECC is enabled the CRC uses the corrected data. If ECC is not enabled (either bypass or simple parity) the CRC uses the uncorrected data. ECC errors, either SEC or DED, are not reported to the system for CRC reads. CRC reads of the Flash also do not affect the FLT features or SFR bits.

##### 29.2.5.6.1. CRC Example Configurations

The following table shows an example CRC configuration with a polynomial size of 32. For each term ($x^n$) of the polynomial used, a corresponding bit in the CRCPOLY Register is set to '1'. This polynomial implements CRC-32-IEEE 802.3.

**Table 29-3.** Example 32-Bit Polynomial CRC Setup

| Field Name | Bit Value | Description |
|---|---|---|
| CRCMODE.PLEN32 | 1 | Polynomial Length is 32-bits |
| CRCPOLY | 0000 0100 1100 0001 0001 1101 1011 011- | CRC polynomial: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ |

The following table shows an example CRC configuration with the polynomial size of 16. The value of CRCPOLY shows the 12th bit and the 5th bit set to '1', as required by the equation. The 0th bit required by the equation is always XOR'd.

**Table 29-4.** Example 16-Bit Polynomial CRC Setup

| Field Name | Bit Value | Description |
|---|---|---|
| CRCMODE.PLEN32 | 0 | Polynomial Length is 16-bits |
| CRCPOLY | 0000 0000 0000 0000 0001 0000 0010 000- | CRC polynomial: $x^{16} + x^{12} + x^5 + 1$ |

#### 29.2.5.7. Wait States Used by CRC

CRC always uses system wait states.

#### 29.2.5.8. Effects of Programming on CRC

The CRC logic pauses automatically when a programming or erase operation prevents it from reading the Flash.

#### 29.2.5.9. Effects of DBGCTRL on CRC

When DBGCTRL.CRCRUN = 1 the CRC Logic continues operation during Debug Mode. When DBGCTRL.CRCRUN = 0 the CRC Logic halts operation during Debug Mode.


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

#### 29.2.5.10. Effects of Sleep Mode on CRC

When RUNSTDBY = 1 and STATUS.PRM=0 (Flash Auto Standby) the CRC Logic continues operation during Standby mode. Otherwise, CRC logic halts operations. When RUNSTDBY = 0 OR STATUS.PRM=1 (Flash Hibernate) the CRC Logic halts operation during Standby mode. Unlike PAUSE, the counters also stop.

**Note:** The PRM status can change on entry to Standby based on CTRLB.SLP[]. The CRC logic uses the state of PRM when the device is in Standby to determine if the CRC runs or not.

#### 29.2.5.11. CRCCTRL – CRC CONTROL REGISTER

##### 29.2.5.11.1. RUNSTDBY

RUNSTDBY determines if the CRC will be calculated when in Standby Mode. When RUNSTDBY = 0, the CRC is not calculated. CRC can only be calculated in Standby if STATUS.PRM = 0.

##### 29.2.5.11.2. CRCEN

CRCEN is essentially the “enable” and “go” bit. All other CRC related SFR bits and fields must be setup prior to writing CRCEN to one. When CRCEN = 1 all other CRC bits are write protected except CRCEN, CRCRST, and PAUSE.CRCEN auto clears when the operation finishes (CRCDONE==1).

##### 29.2.5.11.3. CRCRST

CRCRST resets the CRC SFR bits and returns the FSM to its idle state. The FSM clears CRCRST before returning to idle.

#### 29.2.5.12. CRCCTRL SFR Description

##### 29.2.5.12.1. PERIOD[11:0]

The PERIOD field defines the number of PerCLK counts that the CRC FSM waits from the last bit shifted in to the LFSR until the next read from flash. The request to count is synchronized between the PerCLK domain and the System clock domain. Therefore up to two PerCLK and two System clock periods are added to the period.

##### 29.2.5.12.2. RIN

The RIN, Reflected Input, controls the shift direction of each byte. When RIN = 0, the FSM feeds the LSFR LSbit first. When RIN = 1, the MSbit is first into the LFSR. RIN does not swap bytes in a (32-bit) word or a Flash word, bytes flow from lowest addressed to highest addressed.

##### 29.2.5.12.3. ROUT

The ROUT, Reflected Output, controls the bit order output of the CRC Accumulator. When ROUT = 0, the FSM XORS the CRCACC with CRCFXOR directly. When ROUT = 1, the FSM reverses the bit order of the CRCACC before it XORS it with CRCFXOR.

##### 29.2.5.12.4. AUTOR

The AUTOR selects between Manual mode and Auto Repeat mode. Manual mode requires software to start the CRC engine for each check of the message allowing. Manual mode allows software to check several different messages. Auto Repeat mode does not require software to restart the CRC engine to check the message. However, this mode is limited to checking the same message repeatedly, only stopping on an error or by software disable of the CRC.

##### 29.2.5.12.5. PLEN32

The PLEN32 bit determines the Polynomial Length, either CRC-16 (when zero) or CRC-32 (when one). PLEN32 also defines the effective length of the bits in CRCIV, CRCACC, CRCPOLY, CRCFXOR, and CRCSUM.

Other CRC functions may have a fully programmable PLEN from 0 to 31. These values define CRC lengths from 1 to 32. So PLEN32=0 and PLEN32=1 are equivalent to PLEN=15 and PLEN=31, respectively, on a fully programmable CRC.


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

#### 29.2.5.13. CRCMADR SFR Description

CRCMADR is the byte address of the first byte in the message to be checked. The CRC logic requires a physical system address to determine the region of Flash targeted: boot vs main.

#### 29.2.5.14. CRCMLEN SFR Description

The CRCMLEN field defines the message length in bytes. The Message Length is limited to the size of the largest contiguous region of Flash memory.

#### 29.2.5.15. CRCIV SFR Description

The CRCIV defines the Initial Value which is loaded into the CRCACC at the beginning of the CRC calculation. Both Manual and Auto Repeat modes use the CRCMODE.PLEN32 control to determine the Initial Value size. When selecting CRC-16, bit 16 and above are not used.

#### 29.2.5.16. CRCACC SFR Description

The CRCACC, CRC Accumulator, reports the LFSR state which changes during the calculation for each byte shifted in. Section CRCSUM SFR Description defines how the compare equation uses CRCACC when the CRC calculation completes. CRCMODE.PLEN32 controls the accumulator size. When selecting CRC-16, bit 16 and above are not used.

#### 29.2.5.17. CRCPOLY SFR Description

The CRCPOLY holds the desired Polynomial for the CRC. The highest bit of the polynomial is not contained in CRCPOLY but is always set to one in the CRC logic. The lowest bit, CRCPOLY[0], is always one and not writable. PLEN32 controls the polynomial size. When selecting CRC-16, bit 16 and above are not used.

#### 29.2.5.18. CRCFXOR SFR Description

CRCFXOR contains the value to XOR with CRCACC in the final calculation before comparing to CRCSUM. Section CRCSUM SFR Description defines how the compare equation uses CRCFXOR when the CRC calculation completes. CRCMODE.PLEN32 controls the Final XOR size. When selecting CRC-16, bit 16 and above are not used.

#### 29.2.5.19. CRCSUM SFR Description

CRCSUM contains the checksum value. CRCCTRL.PLEN32 controls the checksum size. When selecting CRC-16, bit 16 and above are not used.

**The CRC Compare Equation is given by:**

Compare is True if CRCSUM = 1.

CRCSUM = (CRCACC xor CRCFXOR)

**Results**

Recall as stated in the Overview, that if the comparison fails (CRCSUM = 0), logic sets the CRCERR flag and stops the process by setting the DONE flag.

If no error occurs (CRCSUM = 0):

If AUTOR is not set (calculations only perform once), the CRC logic sets CRCDONE and clears CRCERR. No further checking happens until CRCDONE is cleared and CRCEN is set to load the initial values.

If AUTOR is set (calculations repeat), then the CRC logic reloads all initial settings and restarts the CRC calculation for the next value without setting CRCDONE. Thus CRC checking will be continuously performed until either an error is encountered, or CRCEN is cleared.

#### 29.2.5.20. CRC Bits in INTFLAG

#### 29.2.5.20.1. CRCDONE

For manual CRC, the CRCDONE bit indicates that the operation is complete with all bits in the message acted upon. The CRCERR bit may, or may not, be set. For auto repeat CRC, the CRCDONE


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

bit is only set when there is an error so CRCERR is also set. This condition also causes the looping to stop.

#### 29.2.5.20.2. CRCERR

The CRCERR bit reports the status of the compare at the end of the CRC when all bits in the message finish shifting into the LFSR.

Recall that CRCSUM = (CRCACC xor CRCFXOR. The CRC Error then becomes:

CRCERR = !CRCSUM

### 29.2.6. Error Correction Logic

There are two device configuration fields (i.e. fuses) stored in the Flash that are used for ECC control. The first is FECCCTL which is the reset value of the SFR based ECCCTRL.ECCCTL[1:0]. The second is FECCUNLCK which is the reset value of ECCCTRL.ECCUNLCK.

#### 29.2.6.1. ECCCTRL SFR Description

#### 29.2.6.1.1. ECCUNLCK

The ECC mode of the Flash can be locked for the duration of the program lifetime in Flash. When FECCUNLCK is 0, ECCUNLCK is also 0 and the selected ECC mode cannot be changed until Flash is updated. This option prevents undesired changing of the mode. When FECCUNLCK is 1 (the default erased state of the flash), ECCUNLCK and ECCCTL can be modified.

**Note:** If ECCUNLCK is 0, debug mode cannot override the ECC or error reporting via DBGCTRL.

#### 29.2.6.1.2. ECCCTRL - 128/256-bit

The field ECCCTL determines how the parity bits are used for Flash reads and writes. The four Modes - ECC Strict, Dynamic, ECC Write w/Dynamic Error and Bypass - affect reads and writes differently.

For the above modes, writes to the flash also write ECC Control Bits, CTL[2:0], which store whether ECC or Simple Parity was calculated on the data. Control Bits exists for each flash word (128-bit or 256-bit data). If the FCW performs a Single Write then the CTL is written with 3'b111 (i.e. not changed from the default erase value of the bits) for Parity. If the FCW performs a Quad Write, the CTL is written as 3'b000 for ECC. CTL[2:0] must be 3'b111 for Single Writes using Simple Parity since all CTL[] bits are not updated with a Single Write. CTL[2:0] is updated for Quad Writes so 3'b000 works for selecting ECC.

The ECC Control Bits table provides the allowed operations based on ECCCTL and its effect on the CTL[2:0] field.

#### 29.2.6.1.3. Strict ECC Writes with ECC Reads - 128/256-bit

If ECCCTL[1:0]= 2'b00, ECC hardware is always active. This mode is also referred to as ECC On.

Only Quad Write programming is allowed in this mode, which always calculates and stores ECC values for the data. The FCW disables the Single Write Program command by making it a no-op if an attempt is made to execute it. Note Row Write uses multiple Quad Writes for programming.

Reads in this mode ignore Flash ECC Control bits CTL[2:0] and always perform ECC error checking and correction. This mode can generate ECC SEC events or ECC DED events.

**Note:** Reads of CFM uses Dynamic to support. The FCR forces ECC to be enabled for all calibration word reads during the reset sequence.

#### 29.2.6.1.4. Dynamic Writes with Dynamic Reads - 128/256-bit

If ECCCTL[1:0]= 2'b01, ECC hardware dynamically switches between ECC and Parity. This mode is also referred to as Dynamic ECC.

Both Quad Write and Single Write programming are allowed in this mode.


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

Reads in this mode obey Flash ECC Control bits CTL[2:0] to determine whether ECC or Parity is calculated on the data. This mode can generate ECC SEC events, and ECC DED interrupt, and Parity Errors that are reported via the ECC DED interrupt.

Note that Flash ECC Control bits CTL[2:0] uses a majority detect during reads to make it tolerant to single bit failures. A single bit failure of CTL[2:0] is reported via the ECC SEC interrupt, and only occurs in this mode. If an ECC SEC occurs at the same time as a CTL SEC, the error is reported on ECC SEC interrupt since both single bit failures get corrected. They also only count as one error with respect to SECCNT.

#### 29.2.6.1.5. ECC Writes with Dynamic Reads - 128/256-bit

If ECCCTL[1:0]= 2'b10, ECC hardware dynamically switches between ECC and Parity. However, only Quad Writes are allowed. Neither DED nor Parity Errors cause a Bus Error. The DERR bit is still set for uncorrectable errors. Otherwise, this mode behaves the same as in "Dynamic Writes with Dynamic Reads", previously.

#### 29.2.6.1.6. Dynamic Writes with No Error Checks - 128/256-bit

If ECCCTL[1:0] = 2'b11, ECC hardware is bypassed (for reads). This mode is also referred to as ECC Bypass.

Both Quad Write and Single Write programming are allowed in this mode.

Reads in this mode ignore Flash ECC Control bits CTL[2:0] and never check ECC or Parity. This mode never generates SEC, DED, or single word parity bus errors.

### 29.2.6.2. ECC Double Error Detected (FFLTSYN.DSERR)

When Read w/ ECC or Read w/Parity is active (i.e. not OFF/Bypass) and a read from flash memory results in an ECC DED, the FCR returns a bus error to the initiator. The error is returned in-band with the data and the initiator determines how to handle the corrupted data.

For the ECC DED, read data is loaded into the read buffer, prefetch buffer, or cache (depending on implementation) with a DED tag so subsequent reads of the address also cause bus errors. In the case of a prefetch, the error is only reported if the address is actually read.

**Note:** SERR and DERR are now combined into a single value (DSERR) tracking both errors. SERR now refers to the Flash SEC Interrupt Clear Enable value INTENCLR.SERR. However SERR and DERR may still be referenced in the datasheet as pertaining to Single or Double Errors.

### 29.2.6.3. Single Error Corrected (FFLTSYN.DSERR)

#### 29.2.6.3.1.

When ECC is active (i.e. not OFF/Bypass) and a read from Flash memory results in an ECC Single Error Corrected (SEC), the FCR reports it via an interrupt (when SECCNT=0) since it is not a critical error. Data in the read buffer, prefetch buffer, or cache (depending on implementation) is correct and no further ECC events are generated for reads that hit the buffer as long as that data is in the buffer. See Section 2.5 for SEC on access to VSS pages for a device family that has an HSM option.

Each read of the flash that results in an SEC causes the FCR to set FFLTSYN.DSERR when SECCNT[] = 0. If the count in SECCNT is non-zero the FCR decrements it and does not set DSERR. The FCR does not reload SECCNT[] when it is zero. Software must write the desired count each time it services the SEC interrupt.

When Dynamic ECC is active, an SEC can be caused by a bit error in CTL. If this is the case, a DED and an SEC can be caused by the same read.

**Note:** SERR and DERR are now combined into a single value (DSERR) tracking both errors. SERR now refers to the Flash SEC Interrupt Clear Enable value INTENCLR.SERR. However SERR and DERR may still be referenced in the datasheet as pertaining to Single or Double Errors.


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

#### 29.2.6.4. Simple Parity Error - 128/256-bit

When Dynamic ECC is active and the read is from an address where CTL selects simple parity, a parity error causes the same behavior as an ECC DED. The error is reported in-band (if ECCCTL[1:0]= 2'b01) and the FFLTSYN.DSERR bit is set.

Unlike a DED, a Bus Error exception is not returned to the initiator.

#### 29.2.6.5. ECC Control Bits

**Table 29-5.** ECC Control Bits

| Error Correction Mode | ECCCTL[1:0] | Write Value CTL[2:0] | Read Values CTL[2:0] | Operation |
|---|---|---|---|---|
| Bypass Mode | 2'b11 | 3'b000 | Ignore | Quad Write w/ ECC<br>Read w/o ECC Check |
| | | 3'b111 | Ignore | Single Write w/ Parity<br>Read w/o Parity Check |
| ECC Write Dynamic Read Mode | 2'b10 | 3'b000 | 3'b000<br>3'b001<br>3'b010<br>3'b100 | Quad Write w/ ECC<br>Read w/ ECC |
| | | 3'b111 | 3'b111<br>3'b110<br>3'b101<br>3'b011 | Single Write Not Available<br>Read w/ Parity |
| Dynamic Mode | 2'b01 | 3'b000 | 3'b000<br>3'b001<br>3'b010<br>3'b100 | Quad Write w/ ECC<br>Read w/ ECC |
| | | 3'b111 | 3'b111<br>3'b110<br>3'b101<br>3'b011 | Single Write w/ Parity<br>Read w/ Parity |
| ECC Strict Mode | 2'b00 | 3'b000 | Ignore | Quad Write w/ ECC<br>Read w/ ECC |
| | | N/A | N/A | Single Write is not available<br>All reads use ECC. |

**Note:** If switching modes, Single Writes in Bypass or Dynamic cause SEC and DED errors in ECC mode. It is highly recommend to select and use only one Error Correction Mode.

### 29.2.7. ECC Fault Control

The flash system contains a fault injection mechanism to allow periodic testing of the ECC logic and error event generation. It can be setup to test Parity, ECC SEC, and ECC DED data faults. For Dynamic mode, it can also inject faults in the ECC Control Bits. The registers ECCCTRL, FFLTCTRL, FFLTPTR, and FFLTADR provide control, while FFLTCAP, FFLTPAR and FFLTSYN provide status. Flash Fault registers are named FFLT* to distinguish them for other memory fault registers.

**Note:** If the system level ECCCTL bits are set for Bypass mode, the fault logic behaves as if it is in Dynamic Mode so that the functional safety code can be developed without generating system level errors.

The ECC Fault logic has two modes of operation: Injection and Capture. Injection mode allows user code to test the ECC logic by forcing faults into the Flash read or write paths. Fault injection behaves


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

slightly differently depending on the ECCCTL mode. Capture mode allows user code to monitor read faults by capturing the first address at which an ECC or Parity error occurs.

### 29.2.7.1. Flash Panel ECC Organization

The following table shows the ECC Calculation Vector bit order with respect to flash data bits and parity bits. The first column is the ECC Calculation Vector. The next two columns map the Read/Write Data and ECC Parity bits to the Calculation Vector. V[n] defines the bit which is selected by FFLT*PTR, so n=FFLT*PTR.

**Table 29-6.** Flash ECC Vector

| Vector Bits | Data Bits | ECC Parity Bits |
|---|---|---|
| V[n] | D[n] | EP[n] |
| 0 | - | 0 |
| 1 | - | 1 |
| 2 | - | 2 |
| 3 | 0 | - |
| 4 | - | 3 |
| 7:5 | 3:1 | - |
| 8 | - | 4 |
| 15:9 | 10:4 | - |
| 16 | - | 5 |
| 31:17 | 25:11 | - |
| 32 | - | 6 |
| 63:33 | 56:26 | - |
| 64 | - | 7 |
| 71:65 | 63:57 | - |
| 127:72 | 119:63 | - |
| 128 | - | 8 |
| 136:129 | 127:120 | - |
| 255:137 | 246:128 | - |
| 256 | - | 9 |
| 265:257 | 255:247 | - |

The following table shows the ECCCTL and CTL field decode. The CTL field is stored in Flash and the fault logic can inject errors into it based on FFLTCTRL.CTLFLT.

### 29.2.7.2. Usage Model for Fault Injection in ECC Mode

Fault Injection occurs at only the Flash address selected by the SFR FFLTADR[31:0]. The FFLTCTRL field FLTMD[2:0] determines the type of fault injected, (single or double) and if it is for reads or writes of Flash. The two fields in the FFLTPTR register, FLT1PTR and FLT2PTR, point to the Vector bit or bits (as ordered in the Flash ECC Vector Table) to invert. Single fault injection always uses FLT1PTR.

Fault Injection always occurs between the ECC logic and the Flash. For writes this means that errors are injected after the ECC/Parity calculation but prior to the data write to Flash. For reads this means that errors are injected after the read from Flash but prior to the ECC/Parity calculation.

In FFLTPAR, the SFR fields SECIN and DEDIN capture the parity bits as they are read from the Flash. They have no meaning for writes as the bits are always driven to zeros (during reads and writes) for the calculation. The SFR fields SECOUT and DEDOUT capture the calculated value for either reads or writes.

In FFLTSYN, the SFR field SECSYN captures the syndrome of the read. This is the XOR of the SECIN with SECOUT. The SFR field DEDSYN captures the overall parity of the values read from Flash. If


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

there is NO Overall Parity change this bit is ZERO. If there is an Overall Parity change this bit is ONE. The following table shows how each of the four conditions are determined. For SEC, the SECSYN points to the bit that was in ERROR.

Though the ECC/Parity mode bits CTL are not used in this mode, the CTLFLT bits are still effective. They will alter writes and reads of that field. However, since CTL is not used in ECC STRICT Mode, errors in CTL have no effect on the ECC calculation or correction.

In ECC Strict Mode, the fields PERR, CTLSTAT and CERR in FFLTSYN are meaningless.

**Table 29-7.** Error Decode

| SECSYN | DEDSYN | SERR | DERR | Condition |
|---|---|---|---|---|
| Zero | 0 | 0 | 0 | No Error |
| Zero | 1 | 1 | 0 | DED parity Bit Error |
| Non-Zero | 0 | 0 | 1 | Double Error |
| Non-Zero | 1 | 1 | 0 | SEC – Data bit error corrected |

#### 29.2.7.3. Usage Model for Fault Injection in Dynamic Model

When ECCCTRL.ECCCTL selects ECC Strict or Dynamic mode, the FFLTCTRL.CTLFLT bits can also be used to inject a fault in the Dynamic ECC CTL bits. This operation is performed in parallel with data fault injection.

Though more than one CTL bit can be inverted, the logic is only capable of correcting single bit errors in the CTL field. For writes, the selected bit is stored in Flash inverted from the normal value. For reads the selected bit is inverted before it is used to determine the ECC mode.

The CTLSTAT field in FFLTSYN reports the captured values of CTL from the read or write that hit FFLTADR. The ECC Control Bits table shows the values of CTL for which the Flash system interprets ECC vs Parity reads and generates the values for writes. Note that injecting an error on the CTL value for a write does not change whether the write uses ECC or Simple Parity.

When performing a Quad Write operation the fault logic, control and status behave the same as in ECC Mode. When performing a read and FFLTSYN.CTLSTAT indicates “Read w/ ECC”, the fault logic, control and status also behave as in ECC Mode. However, FFLTSYN.CERR indicates if CTLSTAT has a bit failure.

When performing Single Write operations, the fault logic injects faults the same way as in ECC Mode. However, some parity bits are not used for Simple Parity (see the ECC Control Bits table) and do not affect the calculation when read. Also, the SEC* and DED* fields plus PERR, CLSTAT, CERR, DERR and SERR are “Don’t Care”.

When performing a read and FFLTSYN.CTLSTAT indicates “Read w/ Parity”, SEC*, DED*, DERR, SERR are “Don’t Care”. The value in FFLTSYN.PERR reflects the error status of each Write Word. If a PERR = 1, then a bus error is generated on the DED error path.

#### 29.2.7.4. Usage Model for Fault Capture

When FFLTCTRL.FLTMD selects Fault Capture Mode, logic captures ECC/Parity status information. Capture only occurs when the INTFLAG.FLTCAP = 0, and never when it is ONE. In other words, once a fault capture has occurred any additional faults are ignored until INTFLAG.FLTCAP is cleared. Capture occurs for a read only on the detection of a fault which also cause logic to set INTFLAG.FLTCAP = 1. Capture occurs for each flash write but the INTFLAG.FLTCAP bit remains ZERO. This allows write ECC logic to be checked without interfering with fault events causing a read capture. Note that the write ECC/Parity logic is incapable of generating an error interrupt.

If there are simultaneous read errors, the error associated with AHB0 is captured. If AHB0 is not involved, then the error associated with panel 0 is captured and the error associated with panel 1 is dropped. If the simultaneous capture is between a read and a write, the read error is captured.


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

Only fields pertinent to the access type (ECC or Simple Parity) are valid. FFLTCAP.FLTADR always captures the address at which the fault occurs (or for any write). CTLSTAT and CERR are valid for a DYNAMIC reads. If the access uses ECC then control logic updates SECSYN, DEDSYN, SERR, DERR, and all fields in FFLTPAR. If the access uses Simple Parity then control logic updates PERR.

#### 29.2.7.5. FFLTCTRL SFR Description

##### 29.2.7.5.1. FLTEN

When FLTEN = 0, the Fault Injection Logic is disabled and user code can safely update the SFR control bits.

When FLTEN = 1 the Fault Injection Logic performs the operation selected by FLTMD. User code can only modify FFLTCTRL when FLTEN = 1. A write any other Fault SFR results in a bus error.

The operation description of all subsequent bits are predicated on the Fault Logic being enabled.

##### 29.2.7.5.2. CTLFLT

When FLTMD selects one of the Fault Injection modes, CTLFLT[2:0] controls fault injection on the Flash ECC Control bits CTL[2:0].

When CTLFLT[n] = 1 logic injects a fault in the associated bit position of CTL[n].

When CTLFLT[n] = 0 logic does not inject a fault on to the associated bit in CTL[n].

Control faults cannot be injected separately from parity faults. In all other FLTMD modes, CTLFLT[2:0] is unused. Also, CTLFLT is only meaningful for Dynamic ECC mode.

##### 29.2.7.5.3. FLTMD

There are four basic fault modes: Disabled, Capture, Read Fault, and Write Fault. The selected mode takes affect when FLTEN = 1.

Fault Injection can be disabled, and write protected to ensure a fault is never injected in to the flash read or write path.

Fault Capture Mode allows the application to monitor the Flash for errors and take the appropriate action. Often times bit errors detected by ECC SEC are soft failures. They may be retention failures or read disturb failures which are correctable by re-writing the data. An application that wishes to decrease the likelihood of an uncorrectable ECC DED fault can reserve a spare page in Flash. The spare page provides a location to copy the offending page while correcting the SEC bit error(s). The offending page would then be erased and re-written with the original data intact. Fault Capture has a lock mode to prevent accidentally switching to Fault Injection Mode.

Fault Injection mode allows user code to periodically test the ECC logic to verify it is working correctly. There are four options. The first two inject a single or double fault for reads. The second two inject a single or double fault for writes. Depending on the fault testing requirements the write option may not be need for a single panel system since reads and writes share the same ECC calculation module. However, dual panel systems with Live Update never share the same ECC module.

#### 29.2.7.6. FFLTPTR SFR Description

##### 29.2.7.6.1. FLT1PTR

The Fault Pointer inverts the selected bit in the ECC Calculation Vector, V. For Single or Double Fault injection, logic inverts bit V[FLT1PTR], as defined in the Flash ECC Vector table. The "FLASH ECC VECTOR" table shows the mapping between the calculation vector and Data, ECC and Parity bits.

##### 29.2.7.6.2. FLT2PTR

The Fault Pointer inverts the selected bit in the ECC Calculation Vector. For Double Fault injection, logic inverts bit V[FLT2PTR], as defined in the Flash ECC Vector table. The "FLASH ECC VECTOR" table also shows the mapping between the calculation vector and Data, ECC and Parity bits.


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

#### 29.2.7.7. FFLTADR SFR Description

FFLTADR.FLTADR contains the System Physical Flash Address at which a Fault Injection occurs.

#### 29.2.7.8. FFLTCAP SFR Description

FFLTCAP.FLTADR contains the System Physical Flash Address at which a Fault Capture occurred.

#### 29.2.7.9. FFLTPAR SFR Description

This register contains the values input to and output from the ECC Calculation Module. For reads the values are post fault injection but for writes they are pre-fault injection. This register does not contain valid information when using Simple Parity.

**Note:** In Capture Mode faults are not injected.

##### 29.2.7.9.1. SECIN

This is the SEC Parity for data read from the Flash panel. For a write this field is always ZERO. For a 128-bit data Flash SECIN[8] is unused.

##### 29.2.7.9.2. DEDIN

This is the Overall Parity for data read from the Flash panel. For a write this field is always ZERO.

##### 29.2.7.9.3. SECOUT

This is the SEC Parity calculated on either the read or write data.

##### 29.2.7.9.4. DEDOUT

This is the Overall Parity calculated on either the read or write data.

#### 29.2.7.10. FFLTSYN SFR Description

This register contains Simple Parity and ECC Parity status. For reads the values are post fault injection but for writes they are pre-fault injection.

**Note:** In Capture Mode faults are not injected.

##### 29.2.7.10.1. SECSYN

SEC Syndrome represents the bitwise XOR of SECIN and SECOUT. SECSYN is not valid for any write or Simple Parity mode.

##### 29.2.7.10.2. DEDSYN

DED Syndrome represents the overall parity from all data bits, SEC Parity bits and the DED Parity bit read from Flash. DEDSYN is not valid for any write or Simple Parity mode.

##### 29.2.7.10.3. DERR and SERR

Double Error and Single Error status indicate the read state of the data: No Error, Single Error, or Double Error. The Error Decode table shows how DEDSYN and SECSYN determine their values. These bits are not valid for any write or Simple Parity mode.

##### 29.2.7.10.4. CERR

Control Error indicates the single bit error status of the CTL[2:0] field. CERR is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

##### 29.2.7.10.5. CTLSTAT

CTLSTAT[2:0] reports the read value of the CTL [2:0] field stored in Flash for the addressed (FLTADR) Flash word (128-bit or 256-bit data). CTLSTAT is only meaningful for reads in Dynamic ECC or Bypass ECC modes.

##### 29.2.7.10.6. PERR

Parity Error captures the difference between each word's read parity and its calculated parity. The word size is dependent on the Flash and is either 32-bit or 64-bit. PERR[0] represents the error status of the lowest addressed word in of the Flash read, while PERR[3] is for the highest addressed word. PERR is only meaningful for reads in Dynamic ECC or Bypass ECC modes.


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

### 29.2.8. Flash Read Protect Features

Flash read protection features are for the security of protected information. There are two types of read protection:

1. Reads to a page locked via CRP and cannot be read until the next reset. This typically refers to boot code including those done by the Boot ROM.
2. Device families that have an HSM option. Note that reads to a VSS page can only be accessed by the HSM.

Read protection controlled by the Flash system is directly protected within Flash system and does not require setup external to the FCR.

#### 29.2.8.1. CFG Read Protect (CRP)

CRP *RP bits provide read protection for CFM pages. Some CFM pages may contain secret information accessible only to the Boot ROM code. The Boot ROM can protect and lock these page before it exits. The BootCfg page contains values to be used by the Boot ROM to program the CRP register before allowing boot flash to execute. See the device spec for this location do so. When read protected, read of this page cause a bus error.

If TrustZone is enabled on the device, then pages not used by the Boot ROM could be left unlocked such that trusted code has the option to select the page settings. However, if the device is not a TrustZone device then the Boot ROM should lock the protection setting while keeping the page readable.

When read protected, reads of this space generate a bus error, return data of all 1's and do not report SEC/DED ECC or Parity errors. Internal sources such as CRC and FLT are blocked (via return data of all 1's) w/o generation of an access error.

##### 29.2.8.1.1. Local Lock Bits

The user may optionally select to prevent further writes to a CRP register bits by also setting associated local LOCK bits (in the same register) when writing the CRP value.

When a local lock bit is set (BC*n*RPLOCK = 1, *n* = 1 or 2), subsequent writes to that register (even if the unlock sequence is followed) have no effect, creating a "write once' register. Local LOCK bits revert to the unlocked state at reset.

## 29.3. Peripheral Dependencies

**Table 29-8.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| FCW | 0x44004000 | 1 : FCW | **AHB:** CLKMSK[0] MASK2 | --- | 2 STATUS0 PERID2 | | | PD_CORE_SW |


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

## 29.4. Peripheral Dependencies

**Table 29-9.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| FCR | 0x44002000 | 0 : FCR | **AHB:**<br>CLKMSK[0]<br>MASK1 | --- | 1<br>STATUS0<br>PERID1 | | | PD_CORE_SW |


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

## 29.5. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | PRIV | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CTRLOP | 7:0 | PREPG | | | | NVMOP[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 ... 0x0B | Reserved | | | | | | | | | |
| 0x0C | INTENCLR | 7:0 | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| | | 15:8 | | | WRERR | RSTERR | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x10 | INTENSET | 7:0 | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| | | 15:8 | | | WRERR | RSTERR | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | INTFLAG | 7:0 | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| | | 15:8 | | | WRERR | RSTERR | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | INTFLAGSET | 7:0 | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| | | 15:8 | | | WRERR | RSTERR | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | STATUS | 7:0 | | | | | | | | BUSY |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | KEY | 7:0 | FEATURE[7:0] | | | | | | | |
| | | 15:8 | CODE[7:0] | | | | | | | |
| | | 23:16 | CODE[15:8] | | | | | | | |
| | | 31:24 | CODE[23:16] | | | | | | | |
| 0x24 | ADDR | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x28 | SRCADDR | 7:0 | SRCADDR[7:0] | | | | | | | |
| | | 15:8 | SRCADDR[15:8] | | | | | | | |
| | | 23:16 | SRCADDR[23:16] | | | | | | | |
| | | 31:24 | SRCADDR[31:24] | | | | | | | |
| 0x2C | DATA0 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x30 | DATA1 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x34 | DATA2 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x38 | DATA3 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[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 | DATA4 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x40 | DATA5 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x44 | DATA6 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x48 | DATA7 | 7:0 | DATA[7:0] | | | | | | | |
| | | 15:8 | DATA[15:8] | | | | | | | |
| | | 23:16 | DATA[23:16] | | | | | | | |
| | | 31:24 | DATA[31:24] | | | | | | | |
| 0x4C ... 0x4F | Reserved | | | | | | | | | |
| 0x50 | PWP | 7:0 | PWPSIZE[7:0] | | | | | | | |
| | | 15:8 | PWPEN | PWPLOCK | PWPMIR | | PWPSIZE[11:8] | | | |
| | | 23:16 | PWPBASE[7:0] | | | | | | | |
| | | 31:24 | | | | | PWPBASE[11:8] | | | |


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

### 29.5.1. Control A REGISTER

**Name:** CTRLA
**Offset:** 0x0
**Reset:** 0x00000000
**Property:** R/W

**Table 29-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | PRIV | | |
| Access | | | | | | R/W | | |
| Reset | | | | | | 0 | | |

**Bit 2 – PRIV** Privileged Access Only

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Peripheral registers only accessible in privileged accesses |
| 0 | DISABLE | Peripheral registers accessible in privileged and unprivileged accesses. |


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

### 29.5.2. Control Operation REGISTER

**Name:** CTRLOP
**Offset:** 0x4
**Reset:** 0x00000000
**Property:** R/W

**Table 29-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PREPG | | | | NVMOP[3:0] | | | |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| Reset | 0 | | | | 0 | 0 | 0 | 0 |

**Bit 7 – PREPG** Note: This field can only be modified when BUSY=0 and KEY=<WRKEY Value> - same as NVMOP

**Notes:**

1. This field can only be modified when PAC Write protection is disabled, STATUS.BUSY=0, and KEY=<WRKEY Value> - same as NVMOP below.
2. It is recommended to either always use or never use PREPG.

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Program Operations include Pre-Program step |
| 0 | DISABLE | Program Operations exclude Pre-Program step |

**Bits 3:0 – NVMOP[3:0]** NVM Operations

**Table 29-12.** Boot ROM Accessible Operations

| Value | Name | Description |
|---|---|---|
| 14 | CE | Chip Erase Operation: Erases PFM+SRF, BFM, CFM |

**Table 29-13.**

| Value | Name | Description |
|---|---|---|
| SINGLEWRITE | Single (Word) Program Opeartion | |


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

| | | |
|---|---|---|
| 2 | QUADWRITE | Quad (Word) Program Operation |

**Table 29-14.**

| Value | Name | Description |
|---|---|---|
| ROWWRITE | Row Write Operation: Programs row selected by ADDR | |
| 4 | PAGEERASE | Page Erase Operation: Erases page selected by ADDR |
| 5 | LPFMERASE | Quad (DWord) Program Operation |


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

### 29.5.3. Interrupt Enable Clear REGISTER

**Name:** INTENCLR
**Offset:** 0xc
**Reset:** 0x00000000
**Property:** R/K

**Table 29-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | WRERR | RSTERR | | | | |
| Access | | | R/K | R/K | | | | |
| Reset | | | 0 | 0 | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 13 – WRERR** Write Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Write Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 12 – RSTERR** Reset or Brown Out Detect Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Reset or Brown Out Detect Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |


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

**Bit 7 – SECERR** Security Violation Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Security Violation Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |

**Bit 6 – OPERR** NVMOP Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the NVMOP Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |

**Bit 5 – WPERR** Write Protection Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Write Protection Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |

**Bit 4 – BUSERR** Bus Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the AHB Bus Error During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |

**Bit 3 – FIFOERR** FIFO Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the FIFO Underrun During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |

**Bit 2 – CFGERR** Configuration Error
Read Value:


© 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 disable the Configuration Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |

**Bit 1 – KEYERR** Key Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Key Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |

**Bit 0 – DONE** NVM Operation Done
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the NVM Operation Done as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Name | Description |
|---|---|---|
| 1 | | Interrupt Enabled |
| 0 | | Interrupt Disabled |


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

### 29.5.4. Interrupt Enable Set REGISTER

**Name:** INTENSET
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** R/S

**Table 29-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | WRERR | RSTERR | | | | |
| Access | | | R/S | R/S | | | | |
| Reset | | | 0 | 0 | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| Access | R/S | R/S | R/S | R/S | R/S | R/S | R/S | R/S |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 13 – WRERR** Write Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Write Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 12 – RSTERR** Reset or Brown Out Detect Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Reset or Brown Out Detect Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |


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

**Bit 7 – SECERR** Security Violation Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Security Violation Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 6 – OPERR** NVMOP Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the NVMOP Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 5 – WPERR** Write Protection Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Write Protection Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 4 – BUSERR** Bus Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the AHB Bus Error During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 3 – FIFOERR** FIFO Underrun Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the FIFO Underrun During Row Write as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 2 – CFGERR** Configuration Error
Read Value:


© 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 Configuration Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 1 – KEYERR** Key Error
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Key Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 0 – DONE** NVM Operation Done
Read Value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the NVM Operation Done as an interrupt request.
Reading this bit returns whether this interrupt is enabled (=1 ≥ enabled).
**Note:** This field can only be modified when STATUS.BUSY=0.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |


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

### 29.5.5. Interrupt Flag REGISTER

**Name:** INTFLAG
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** R/K

**Notes:**

1. The interrupt flag bits of this register are set by hardware only.
2. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 29-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | WRERR | RSTERR | | | | |
| Access | | | R/K/HS | R/K/HS | | | | |
| Reset | | | 0 | 0 | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| Access | R/K/HS | R/K/HS | R/K/HS | R/K/HS | R/K/HS | R/K/HS | R/K/HS | R/K/HS |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 13 – WRERR** Write/Erase Error

**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | The Write/Erase sequence did not complete successfully |
| 0 | The Write/Erase sequence completed normally |

**Bit 12 – RSTERR** Reset Error

The error is only captured during Write/Erase operations.
**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

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 interrupt pending.

| Value | Description |
|---|---|
| 1 | A reset or Low Voltage Detected (possible data corruption, verify data) |
| 0 | No Reset and Voltage level OK during write/erase |

**Bit 7 – SECERR** Security Violation Error

Attempted operation violates security configuration.
**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Security Violation Error |
| 0 | No Security Violation Error |

**Bit 6 – OPERR** NVMOP Error

**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Selected Operation is Disabled Error |
| 0 | No NVMOP Error |

**Bit 5 – WPERR** Write Protect Error

**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Write Protection Error |
| 0 | No Write Protection Error |

**Bit 4 – BUSERR** Bus Error

**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Bus Error |
| 0 | No Bus Error |

**Bit 3 – FIFOERR** FIFO Error

**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | FIFO Error |


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

| Value | Description |
|---|---|
| 0 | No FIFO Error |

**Bit 2 – CFGERR** Configuration Error

Attempted Write/Erase when disallowed by a configuration setting.
**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | CFC Error |
| 0 | No CFG Error |

**Bit 1 – KEYERR** Key Error

Attempted to write to an SFR bit without first enabling it via KEY.
**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Key Error |
| 0 | No Key Error |

**Bit 0 – DONE** Done

When NVMOP completes the FSM clears BUSY and sets Done.
**Note:** While this bit and its associated interrupt enable bit are 1, the NVM interrupt remains asserted.

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | NVMOP Done |
| 0 | NVMOP Not Done |


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

### 29.5.6. Interrupt Flag Set REGISTER

**Name:** INTFLAGSET
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** R/S

**Notes:**

1. The interrupt flag bits of this register are set by hardware only.
2. Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 29-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | WRERR | RSTERR | | | | |
| Access | | | R/S/HS | R/S/HS | | | | |
| Reset | | | 0 | 0 | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SECERR | OPERR | WPERR | BUSERR | FIFOERR | CFGERR | KEYERR | DONE |
| Access | R/S/HS | R/S/HS | R/S/HS | R/S/HS | R/S/HS | R/S/HS | R/S/HS | R/S/HS |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 13 – WRERR** Write Error
Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 12 – RSTERR** Reset Error
Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.


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

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 7 – SECERR** Security Error
Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 6 – OPERR** NVMOP Error
Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 5 – WPERR** Write Protect Error
Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 4 – BUSERR** Bus Error
Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 3 – FIFOERR** FIFO Error
Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 2 – CFGERR** Configuration Error
Read value reflects state of INTFLAG.
Write value:


© 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 set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 1 – KEYERR** Key Error

Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 0 – DONE** Done

Read value reflects state of INTFLAG.
Write value:
Writing a '0' to this bit has no effect.
Writing a 1 to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |


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

### 29.5.7. Status REGISTER

**Name:** STATUS
**Offset:** 0x1c
**Reset:** 0x00000000
**Property:** R

**Table 29-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | BUSY |
| Access | | | | | | | | R/HS/HC |
| Reset | | | | | | | | 0 |

**Bit 0 – BUSY** NVM Busy Status

**Note:** This bit is read-only. It is set and cleared by hardware.

| Value | Name | Description |
|---|---|---|
| 1 | BUSY | NVM Busy - All SFR bits are not writable |
| 0 | IDLE | NVM Not Busy |


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

### 29.5.8. Feature Unlock Key REGISTER

**Name:** KEY
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** R/W

**Notes:**

1. KEY can only be written when STATUS.BUSY=0.
2. KEY resets to 0 (a valid lock code) on a successful write to CTRLA, SWAP or an unsuccessful write to any register (any write that sets KEYERR).
3. KEY allows for multiple writes to configuration registers when using CFGKEY. To lock, write a Valid Lock Code to KEY.
4. Byte and Half-Word writes are ignored but result in a bus error.

**Table 29-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CODE[23:16] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | CODE[15:8] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | CODE[7:0] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | FEATURE[7:0] | | | | | | | |
| Access | R/W/HC | R/W/HC | R/W/HC | R/W/HC | R/W/HC | R/W/HC | R/W/HC | R/W/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:8 – CODE[23:0]** Lock Code

This register is used to lock and unlock write access to other FCW SFR bits.
The KEY state remains active until it is written to a Valid Lock Code or CTRLOP is written.
0x91C32C= Valid Unlock code - Unlocks selected SFR(s)
All Others = Valid Lock Code - Locks all SFR protected by KEY register

**Bits 7:0 – FEATURE[7:0]**

This register is used to lock and unlock write access to other FCW SFR bits.
0x00= no selection, invalid selection, or invalid Key - Locks all SFR protected by KEY register
0x01= WRKEY Value: Unlock SFR bits associated with Write/Erase
0x04= CFGKEY Value: Unlock SFR bits associated with general Flash configuration
all other values: Invalid selection - Key Value set to 00 - Locks all SFR protected by KEY register


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

### 29.5.9. Flash Destination Address REGISTER

**Name:** ADDR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** R/W

**Table 29-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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]**

This is a system byte address that the FCW aligns (by dropping lower ordered bits) to the minimum resolution of the NVMOP. For more information on valid operations, see the NVMOP register.

**Note:** This field can only be modified when BUSY=0.


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

## 29.5.10. System Source Address REGISTER

**Name:** SRCADDR
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** R/W

**Table 29-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[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 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[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 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[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 |
|---|---|---|---|---|---|---|---|---|
| | SRCADDR[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 – SRCADDR[31:0]** System Address
This is the system physical address of the data to be programmed into the Flash when CTRLA.NVMOP is set to Row Write SRCADDR[1:0] are always 0.
**Note:**
1. This field can only be modified when STATUS.BUSY=0.


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

### 29.5.11. Flash Write Data n REGISTER

**Name:** DATA
**Offset:** 0x2C + n*0x04 [n=0..7]
**Reset:** 0x00000000
**Property:** R/W

**Table 29-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 value in this register(s) is written to Flash when a Write operation is commanded.
Single Write: (32-bit data)
Writes DATA0 to ADDR[31:2] with address bits [1:0] = 00
Quad Write: (128-bit data)
Writes DATA0 to ADDR[31:4], with address bits[3:0] = 0000
Writes DATA1 to ADDR[31:4], with address bits[3:0] = 0100
Writes DATA2 to ADDR[31:4], with address bits[3:0] = 1000
Writes DATA3 to ADDR[31:4], with address bits[3:0] = 1100
**Note:** This field can only be modified when STATUS.BUSY=0.


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

## 29.5.12. PFM Write Protect Region n REGISTER

**Name:** PWP
**Offset:** 0x050
**Reset:** 0x00000000
**Property:** R/W

**Table 29-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PWPBASE[11:8] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PWPBASE[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 |
|---|---|---|---|---|---|---|---|---|
| | PWPEN | PWPLOCK | PWPMIR | | PWPSIZE[11:8] | | | |
| Access | R/W | R/S | 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 |
|---|---|---|---|---|---|---|---|---|
| | PWPSIZE[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 27:16 – PWPBASE[11:0]** Page Base

The Region Base Address is PWPBASE * 4096.
When PWPEN=1, the region from PWPBASE to PWPBASE+PWPSIZE+1 is write/erase protected.
**Note:** This is a byte address force to align to page boundaries.

**Note:** This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

**Bit 15 – PWPEN** Page Enable

**Note:** This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | PWP is Enabled for the defined region |
| 0 | DISABLE | PWP is Not Enabled for the defined region |

**Bit 14 – PWPLOCK** Page Lock

**Note:** PWPLOCK can be set the same time as PWPBASE and PWPSIZE are written. Once set, PWPLOCK can only be cleared by a reset (Writing zero has no effect.).

**Note:** This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.


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

| Value | Name | Description |
|---|---|---|
| 1 | LOCKED | This register is Locked and cannot be modified |
| 0 | UNLOCKED | This register is Not Locked and can be modified |

**Bit 13 – PWPMIR** Page Mirror

Mirrors Lower PFM settings to Upper or Upper PFM settings to Lower. This feature can be used to maintain Write Protect (WP) consistency between Upper and Lower PFM when using PFSWAP.
**Note:** When Mirrored, the PWPBASE address bit that distinguishes between Upper and Lower PFM is treated as a "Don't Care", meaning it can be 0 or 1.

**Note:** This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.

| Value | Name | Description |
|---|---|---|
| 1 | MIRROR | PWP settings are Mirrored |
| 0 | UNMIRROR | PWP settings are NOT Mirrored |

**Bits 11:0 – PWPSIZE[11:0]** Page Region Size

Pages Region Size is (PWPSIZE+1) *4KB.
0x000= 4KB
0x001= 8KB
...
0x3F= 256KB
**Note:** This field can only be modified when PWPLOCK=0, STATUS.BUSY=0 and KEY.KEY=<CFGKEY Value>.


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

## 29.6. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | PRIV | | |
| | | 15:8 | AUTOWS | ADRWS | | | FWS[3:0] | | | |
| | | 23:16 | | | | | RDBUFWS[3:0] | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CTRLB | 7:0 | CHEINV | | | | | | | PRM |
| | | 15:8 | | | | | | | SLP[1:0] | |
| | | 23:16 | | | | | | | PREFDIS | CHEDIS |
| | | 31:24 | | | | | | | | |
| 0x08 | INTENCLR | 7:0 | | | | | | | | SERR |
| | | 15:8 | | | | | | | CRCERR | CRCDONE |
| | | 23:16 | | | | | | | | FLTCAP |
| | | 31:24 | | | | | | | | |
| 0x0C | INTENSET | 7:0 | | | | | | | | SERR |
| | | 15:8 | | | | | | | CRCERR | CRCDONE |
| | | 23:16 | | | | | | | | FLTCAP |
| | | 31:24 | | | | | | | | |
| 0x10 | INTFLAG | 7:0 | | | | | | | | SERR |
| | | 15:8 | | | | | | | CRCERR | CRCDONE |
| | | 23:16 | | | | | | | | FLTCAP |
| | | 31:24 | | | | | | | | |
| 0x14 | INTFLAGSET | 7:0 | | | | | | | | SERR |
| | | 15:8 | | | | | | | CRCERR | CRCDONE |
| | | 23:16 | | | | | | | | FLTCAP |
| | | 31:24 | | | | | | | | |
| 0x18 | STATUS | 7:0 | | | | | | | | PRM |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | DBGCTRL | 7:0 | | | | | | DBGECC[1:0] | | CRCRUN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | ECCCTRL | 7:0 | | ECCUNLCK | ECCCTL[1:0] | | | | | |
| | | 15:8 | SECCNT[7:0] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | CRCCTRL | 7:0 | | | RUNSTDBY | | | | CRCEN | CRCRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | CRCMODE | 7:0 | | | | | | | | |
| | | 15:8 | RIN | ROUT | AUTOR | PLEN32 | | | | |
| | | 23:16 | PERIOD[7:0] | | | | | | | |
| | | 31:24 | | | | | PERIOD[11:8] | | | |
| 0x2C | CRCPAUSE | 7:0 | | | | | | | | PAUSE |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x30 | CRCMADR | 7:0 | CRCMADR[7:0] | | | | | | | |
| | | 15:8 | CRCMADR[15:8] | | | | | | | |
| | | 23:16 | CRCMADR[23:16] | | | | | | | |
| | | 31:24 | | | | | CRCMADR[27:24] | | | |
| 0x34 | CRCMLEN | 7:0 | CRCMLEN[7:0] | | | | | | | |
| | | 15:8 | CRCMLEN[15:8] | | | | | | | |
| | | 23:16 | CRCMLEN[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 | CRCIV | 7:0 | CRCIV[7:0] | | | | | | | |
| | | 15:8 | CRCIV[15:8] | | | | | | | |
| | | 23:16 | CRCIV[23:16] | | | | | | | |
| | | 31:24 | CRCIV[31:24] | | | | | | | |
| 0x3C | CRCACC | 7:0 | CRCACC[7:0] | | | | | | | |
| | | 15:8 | CRCACC[15:8] | | | | | | | |
| | | 23:16 | CRCACC[23:16] | | | | | | | |
| | | 31:24 | CRCACC[31:24] | | | | | | | |
| 0x40 | CRCPOLY | 7:0 | CRCPOLY[7:0] | | | | | | | |
| | | 15:8 | CRCPOLY[15:8] | | | | | | | |
| | | 23:16 | CRCPOLY[23:16] | | | | | | | |
| | | 31:24 | CRCPOLY[31:24] | | | | | | | |
| 0x44 | CRCFXOR | 7:0 | CRCFXOR[7:0] | | | | | | | |
| | | 15:8 | CRCFXOR[15:8] | | | | | | | |
| | | 23:16 | CRCFXOR[23:16] | | | | | | | |
| | | 31:24 | CRCFXOR[31:24] | | | | | | | |
| 0x48 | CRCSUM | 7:0 | CRCSUM[7:0] | | | | | | | |
| | | 15:8 | CRCSUM[15:8] | | | | | | | |
| | | 23:16 | CRCSUM[23:16] | | | | | | | |
| | | 31:24 | CRCSUM[31:24] | | | | | | | |
| 0x4C | FFLTCTRL | 7:0 | | | | | | | FLTEN | FLTRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x50 | FFLTMODE | 7:0 | | | | | | | | |
| | | 15:8 | | FLTMD[2:0] | | | | CTLFLT[2:0] | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x54 | FFLTPTR | 7:0 | FLTPTR1[7:0] | | | | | | | |
| | | 15:8 | | | | | | | | FLTPTR1[8] |
| | | 23:16 | FLTPTR2[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | FLTPTR2[8] |
| 0x58 | FFLTADR | 7:0 | FLTADR[7:0] | | | | | | | |
| | | 15:8 | FLTADR[15:8] | | | | | | | |
| | | 23:16 | FLTADR[23:16] | | | | | | | |
| | | 31:24 | | | | | FLTADR[27:24] | | | |
| 0x5C | FFLTCAP | 7:0 | FLTADR[7:0] | | | | | | | |
| | | 15:8 | FLTADR[15:8] | | | | | | | |
| | | 23:16 | FLTADR[23:16] | | | | | | | |
| | | 31:24 | | | | | FLTADR[27:24] | | | |
| 0x60 | FFLTPAR | 7:0 | SECIN[7:0] | | | | | | | |
| | | 15:8 | DEDIN | | | | | | | SECIN[8] |
| | | 23:16 | SECOUT[7:0] | | | | | | | |
| | | 31:24 | DEDOUT | | | | | | | SECOUT[8] |
| 0x64 | FFLTSYN | 7:0 | SECSYN[7:0] | | | | | | | |
| | | 15:8 | DEDSYN | | | | | | | SECSYN[8] |
| | | 23:16 | | | | | | CERR | DSERR[1:0] | |
| | | 31:24 | | PERR[3:0] | | | | CTLSTAT[2:0] | | |


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

### 29.6.1. CTRL A Register

**Name:** CTRLA
**Offset:** 0x0
**Reset:** 0x00008000
**Property:** R/W

**Table 29-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | RDBUFWS[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 |
|---|---|---|---|---|---|---|---|---|
| | AUTOWS | ADRWS | | | FWS[3:0] | | | |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 1 | 0 | | | 0 | 0 | 0 | 0 |

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

**Bits 19:16 – RDBUFWS[3:0]** AHB Read Buffer
When returning data from the AHB read buffer, insert wait states to match ADRWS and FWS from Flash based on RDBUFWS.
Read Buffer deterministic access requires AUTOWS=0.

| Value | Name | Description |
|---|---|---|
| 1 | DETERMINISTIC | ADRWS + FWS wait state for hits to AHB read buffer |
| 0 | ZEROWS | 0 Wait States |

**Bit 15 – AUTOWS** Automatic Wait States
Taws = Tacc + 5ns + 2 clocks

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Use Automatic wait states: Total flash wait states are ADRWS + Taws |
| 0 | DISABLE | Use FWS: Total flash wait states are ADRWS + FWS |

**Bit 14 – ADRWS** Address Wait State
For Total flash wait states see AUTOWS.

| Value | Name | Description |
|---|---|---|
| 1 | ONE | Add 1 Address Wait State - allowing for higher clock frequencies |
| 0 | ZERO | Add 0 Address Wait States - allowing for higher performance at lower clock frequencies |


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

**Bits 11:8 – FWS[3:0]** Flash Wait States

**Note:** This is not the wait states seen by the CPU. For Total Flash wait states see AUTOWS.

| Value | Description |
|---|---|
| 15 | Fifteen Wait States |
| 14 | Fourteen Wait States |
| 1 | One Wait State |
| 0 | Zero Wait States |

**Bit 2 – PRIV** Privileged Access Only

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Peripheral registers only accessible in privileged accesses |
| 0 | DISABLE | Peripheral register accessible in privileged and unprivileged accesses. |


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

### 29.6.2. CTRL B Register

**Name:** CTRLB
**Offset:** 0x4
**Reset:** 0x00000000
**Property:** R/W

**Table 29-26.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | PREFDIS | CHEDIS |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | SLP[1:0] | |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CHEINV | | | | | | | PRM |
| Access | R/S/HC | | | | | | | R/S/HC |
| Reset | 0 | | | | | | | 0 |

**Bit 17 – PREFDIS** Prefetch Disable

**When CTRLB.CHEDIS=0:**
1 (DISABLE) = Predictive Prefetch Disabled
0 (ENABLE) = Predictive Prefetch Enabled
**When CTRLB.CHEDIS=1:**
Predictive Prefetch is always disabled and this field is ignored.

| Value | Name | Description |
|---|---|---|
| 1 | DISABLE | Predictive Prefetch Disabled |
| 0 | ENABLE | Predictive Prefetch Enabled |

**Bit 16 – CHEDIS** Cache Disable

| Value | Name | Description |
|---|---|---|
| 1 | DISABLE | Cache Disabled; Interface uses RDBUFWS selection. |
| 0 | ENABLE | Cache Enabled; Accesses to flash use AUTOWS/FSW selection. |

**Bits 9:8 – SLP[1:0]**

**Note:** The status of power reduction is reflected in STATUS.PRM.

| Value | Name | Description |
|---|---|---|
| 3 | AUTOSLOW | Enter Flash Hibernate on entry to Standby w/ wakeup on first access |


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

| Value | Name | Description |
|---|---|---|
| 2 | AUTOFAST | Enter Flash Hibernate on entry to Standby w/ wakeup to Auto Standby |
| 1 | RSVD01 | RSVD |
| 0 | MANUAL | Maintain state of STATUS.PRM on entry to Standby |

**Bit 7 – CHEINV** Cache Invalidate

Cache invalidation takes 1 clock cycle, but may be delayed by an active read.

| Value | Name | Description |
|---|---|---|
| 1 | INV | Invalidate Cache |
| 0 | NONE | No Action |

**Bit 0 – PRM** Flash Power Mode

When written to 1, this bit remains a 1 until STATUS.PRM toggles. Then it is cleared by hardware. Writing a 0 has no effect.

**Note:** The status of power reduction is reflected in STATUS.PRM.

| Value | Name | Description |
|---|---|---|
| 1 | TOGGLE | Toggle the Flash Power Mode |
| 0 | NONE | No Action |


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

### 29.6.3. Interrupt Enable Clear Register

**Name:** INTENCLR
**Offset:** 0x8
**Reset:** 0x00000000
**Property:** R/K

**Table 29-27.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | FLTCAP |
| Access | | | | | | | | R/K |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CRCERR | CRCDONE |
| Access | | | | | | | R/K | R/K |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SERR |
| Access | | | | | | | | R/K |
| Reset | | | | | | | | 0 |

**Bit 16 – FLTCAP**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the ECC Fault Capture as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 9 – CRCERR**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the CRC Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 8 – CRCDONE**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the CRC Calculation Done as an interrupt request.


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

Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 0 – SERR**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will disable the Flash SEC as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |


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

### 29.6.4. Interrupt Enable SET Register

**Name:** INTENSET
**Offset:** 0xc
**Reset:** 0x00000000
**Property:** R/S

**Table 29-28.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | FLTCAP |
| Access | | | | | | | | R/S |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CRCERR | CRCDONE |
| Access | | | | | | | R/S | R/S |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SERR |
| Access | | | | | | | | R/S |
| Reset | | | | | | | | 0 |

**Bit 16 – FLTCAP**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the ECC Fault Capture as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 9 – CRCERR**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the CRC Error as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 8 – CRCDONE**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the CRC Calculation Done as an interrupt request.


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

Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |

**Bit 0 – SERR**

Writing a '0' to this bit has no effect.
Writing a 1 to this bit will enable the Flash SEC as an interrupt request.
Reading this bit returns whether this interrupt is enabled.

| Value | Description |
|---|---|
| 1 | Interrupt Enabled |
| 0 | Interrupt Disabled |


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

### 29.6.5. Interrupt Flag Register

**Name:** INTFLAG
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** R/K

**Table 29-29.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | FLTCAP |
| Access | | | | | | | | R/K/HS |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CRCERR | CRCDONE |
| Access | | | | | | | R/K/HS | R/K/HS |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SERR |
| Access | | | | | | | | R/K/HS |
| Reset | | | | | | | | 0 |

**Bit 16 – FLTCAP**

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | An ECC Fault Capture, related to FLTMD[], occurred |
| 0 | No Interrupt Pending |

**Bit 9 – CRCERR**

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | CRCACC Is Not Equal to the XOR of CRCSUM and CRCFXOR |
| 0 | No Interrupt Pending |

**Bit 8 – CRCDONE**

Reading this bit returns the INTFLAG bit value.
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 interrupt pending.

| Value | Description |
|---|---|
| 1 | Calculation Done |
| 0 | No Interrupt Pending |

**Bit 0 – SERR**

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this disables the pending interrupt.
Writing a '1' to this bit will set the interrupt pending, as if SECCNT count is reached.

| Value | Description |
|---|---|
| 1 | SECCNT Count reached |
| 0 | No Interrupt Pending |


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

### 29.6.6. Interrupt Flag Set Register

**Name:** INTFLAGSET
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** R/S

**Table 29-30.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | FLTCAP |
| Access | | | | | | | | R/S/HS |
| Reset | | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | CRCERR | CRCDONE |
| Access | | | | | | | R/S/HS | R/S/HS |
| Reset | | | | | | | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | SERR |
| Access | | | | | | | | R/S/HS |
| Reset | | | | | | | | 0 |

**Bit 16 – FLTCAP**

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 9 – CRCERR**

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 8 – CRCDONE**

Reading this bit returns the INTFLAG bit value.
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 interrupt pending.

| Value | Description |
|---|---|
| 1 | Set Interrupt Pending |
| 0 | No effect |

**Bit 0 – SERR**

Reading this bit returns the INTFLAG bit value.
Writing a '0' to this disables the pending interrupt.
Writing a '1' to this bit will set the interrupt pending, as if SECCNT count is reached.

| Value | Description |
|---|---|
| 1 | SECCNT Count reached |
| 0 | No Interrupt Pending |


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

### 29.6.7. Status Register

**Name:** STATUS
**Offset:** 0x18
**Reset:** 0xE0000000
**Property:** R

**Table 29-31.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PRM |
| Access | | | | | | | | R/HS/HC |
| Reset | | | | | | | | 0 |

**Bit 0 – PRM** NVM Power Reduction Mode Status

| Value | Description |
|---|---|
| 1 | Flash is Powered Down and wakes on first access (read or NVMOP) |
| 0 | Flash is Ready (for read or NVMOP) |


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

### 29.6.8. Debug Control CTRL Register

**Name:** DBGCTRL
**Offset:** 0x1c
**Reset:** 0x00000000
**Property:** R/W

**Table 29-32.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | DBGECC[1:0] | | CRCRUN |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 2:1 – DBGECC[1:0]** Debug ECC Mode

| Value | Name | Description |
|---|---|---|
| 3 | RSVD11 | Do not use but same as 01 (DISABLE) |
| 2 | ENABLE | Corrected per ECCCTRL; Bus Error, INTFLAG, SEC counter, and FLT logic operates as setup. |
| 1 | DISABLE | Not corrected; No Bus Error; INTFLAG, SEC counter, and FLT logic are not updated. |
| 0 | NOERR | Corrected per ECCCTRL; No Bus ERR; INTFLAG, SEC counter, and FLT logic are not updated. |

**Bit 0 – CRCRUN** CRC Debug Run Enable

| Value | Name | Description |
|---|---|---|
| 1 | RUN | CRC Logic Runs in Debug Mode. |
| 0 | HALT | CRC Logic Halts in Debug Mode |


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

### 29.6.9. ECC Control Register

**Name:** ECCCTRL
**Offset:** 0x20
**Reset:** 0x00000040
**Property:** R/W

**Table 29-33.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | SECCNT[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 |
|---|---|---|---|---|---|---|---|---|
| | | ECCUNLCK | ECCCTL[1:0] | | | | | |
| Access | | R/K | R/W | R/W | | | | |
| Reset | | 1 | 0 | 0 | | | | |

**Bits 15:8 – SECCNT[7:0]**
SECCNT is the start value of an internal counter that decrements by 1 for each occurrence of a SEC event (including ECC CTL[2:0] bit if in Dynamic ECC Mode). The SECCNT counter stops decrementing at zero. If an SEC error occurs when the SECCNT counter is zero, the INTFLAG.SERR flag bit is set.
**Note:** This field counts all SEC errors and is not limited to SEC errors on unique addresses.

**Bit 6 – ECCUNLCK** ECC Unlock
**Note:** This field can only be modified when ECCUNLCK=1.

The read value dictates the unlock state:

| Value | Name | Description |
|---|---|---|
| 1 | UNLOCKED | ECCUNLCK and ECCCTL[1:0] can be written |
| 0 | LOCKED | ECCUNLCK and ECCCTL[1:0] cannot be written |

**Bits 5:4 – ECCCTL[1:0]** ECC CTRL
Condition: {PFM_PNL_DATA_MSB==71} 10 (ECC) = Reserved - Resolves to STRICT 01 (DYNAMIC) = Reserved - Resolves to STRICT
**Note:** This field can only be modified when ECCUNLCK=1.

Restricts one or more NVMOPs:


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

**Table 29-34.**

| Value | Name | Description |
|---|---|---|
| 3 | DISABLE | Dynamic Writes with No Error Check Reads |
| 2 | ECC | ECC Writes with Dynamic Reads |
| 1 | DYNAMIC | Dynamic Writes with Dynamic Reads |
| 0 | STRICT | ECC Writes with ECC Reads |


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

## 29.6.10. CRC Control Register

**Name:** CRCCTRL
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** R/W

**Table 29-35.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | RUNSTDBY | | | | CRCEN | CRCRST |
| Access | | | R/W | | | | R/W/HC | R/S/HC |
| Reset | | | 0 | | | | 0 | 0 |

**Bit 5 – RUNSTDBY** CRC Run in Standby

| Value | Name | Description |
|---|---|---|
| 1 | RUN | CRC Runs in Standby but only if STATUS.PRM=0 (i.e. Flash is in Auto Standby) |
| 0 | HALT | CRC Halts in Standby |

**Bit 1 – CRCEN**

When CRCEN = 1, any write to a CRC registers labeled as Enable-Protected is discarded and an error is generated.
**Note:** When CRCEN = 1 all other CRC* SFR bits are write protected, except CRCEN, CRCRST, and CRCPAUSE.PAUSE

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | CRC Calculation Enabled (w/ start on write to 1). |
| 0 | DISABLE | Stops CRC calculation. |

**Bit 0 – CRCRST** CRC Reset

| Value | Name | Description |
|---|---|---|
| 1 | RESET | Resets all CRC SFR bits. |
| 0 | NONE | No Action |


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

## 29.6.11. CRC Mode Register

**Name:** CRCMODE
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** R/W

**Table 29-36.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PERIOD[11:8] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | PERIOD[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 |
|---|---|---|---|---|---|---|---|---|
| | RIN | ROUT | AUTOR | PLEN32 | | | | |
| Access | R/W | R/W | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 0 | | | | |

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

**Bits 27:16 – PERIOD[11:0]** Period

The number of PerCLK counts CRC logic waits between needing new data and reading that data from Flash. (The PerCLK is clock at an 8 MHz fixed frequency. Non-zero PERIOD values are used to throttle back the bandwidth used for CRC calculations.)
0 = Read Data Immediately
All Other Values = Wait **PERIOD** PerCLK counts + 2 AHB Clocks (for sync) before starting

**Bit 15 – RIN** Reflected Input

This option is sometimes referred to as Reflected Byte or Reflected Input.

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | The LFSR CRC is calculated Least Significant Bit first (Reflected) |
| 0 | DISABLE | The LFSR CRC is calculated Most Significant Bit first (Not Reflected) |

**Bit 14 – ROUT** Reflected Output

This option is sometimes referred to as Reflected Result or Reflected Output.

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | The CRCACC is Reflected (before the Final XOR) |
| 0 | DISABLE | The CRCACC is Not Reflected |


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

**Bit 13 – AUTOR** CRC Auto Repeat

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Continually Repeat CRC calculation; stop on error, set CRCDONE and CRCERR |
| 0 | DISABLE | Perform CRC calculation once then set DONE and if needed, CRCERR. |

**Bit 12 – PLEN32** Polynomial Length Select

| Value | Name | Description |
|---|---|---|
| 1 | THIRTYTWO | Polynomial is 32-bits |
| 0 | SIXTEEN | Polynomial is 16-bits |


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

### 29.6.12. CRC Pause Register

**Name:** CRCPAUSE
**Offset:** 0x2c
**Reset:** 0x00000000
**Property:** R/W

**Table 29-37.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PAUSE |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – PAUSE** Pause Reads

**Note:** The CRC calculation continues until it needs more data, and then it pauses.

Prevent the CRC FSM from reading Flash memory so as to not interfere with CPU activity:

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | Pause CRC Reads of Flash |
| 0 | DISABLE | CRC Reads Flash as Required |


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

### 29.6.13. CRC Message Address Register

**Name:** CRCMADR
**Offset:** 0x30
**Reset:** 0x00000000
**Property:** R/W

**Table 29-38.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CRCMADR[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCMADR[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCMADR[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 27:0 – CRCMADR[27:0]** .

This is the system physical address of the first byte of the message.

**Note:** Not all bits in this register are writable. CRCMADR[27:24] determines which Flash region (CFM, BFM, PFM) is targeted and always exist. For CRCMADR[23:0] only the number of bits required to address the largest flash region (PFM) exists. Bits that do not exist are treated as R-0. The R/W bits of [23:0] match the bits for CRCMLEN.


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

## 29.6.14. CRC Message Length Register

**Name:** CRCMLEN
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** R/W

**Table 29-39.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CRCMLEN[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCMLEN[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCMLEN[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 – CRCMLEN[23:0]**

**Notes:**

1. Only the number of bits needed to address the largest contiguous Flash region are implemented.
2. Software should limit the value written to CRCMLEN such that CRCMADR + CRCMLEN does not exceed the end of the Flash region (i.e. BFM, PFM, or on a secure device between security regions) containing CRCMADR.


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

### 29.6.15. CRC Initial Value Register

**Name:** CRCIV
**Offset:** 0x38
**Reset:** 0x00000000
**Property:** R/W

**Table 29-40.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRCIV[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCIV[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCIV[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCIV[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 – CRCIV[31:0]**

CRCIV is loaded into CRCACC before the start of each CRC calculation, including at the start of repeat loops.

**When CRCMODE.PLEN32=0:**
CRCIV[31:16] is don't care.
CRCIV[15:0] is the initial value.

**When CRCMODE.PLEN32=1:**
CRCIV[31:0] is the initial value.


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

### 29.6.16. CRC Accumulator Register

**Name:** CRCACC
**Offset:** 0x3c
**Reset:** 0x00000000
**Property:** R

**Table 29-41.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRCACC[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCACC[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCACC[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCACC[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CRCACC[31:0]**

When CRCMODE.PLEN32=0: CRCACC[31:16] is don't care. CRCACC[15:0] is the accumulator.
When CRCMODE.PLEN32=1: CRCACC[31:0] is the accumulator.


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

## 29.6.17. CRC Polynomial Register

**Name:** CRCPOLY
**Offset:** 0x40
**Reset:** 0x00000001
**Property:** R/W

**Table 29-42.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRCPOLY[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCPOLY[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCPOLY[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCPOLY[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 31:0 – CRCPOLY[31:0]**

When CRCMODE.PLEN32=0: CRCPOLY[31:16] is don't care. CRCPOLY[15:0] is the polynomial coefficients.
CRCPOLY[0] and CRCPOLY[16] are implied and are always 1.
When CRCMODE.PLEN32=1: CRCPOLY[31:0] is the polynomial coefficients.
CRCPOLY[0] and CRCPOLY[32] are implied and are always 1.

| Value | Description |
|---|---|
| 1 | Enable the XOR input to the shift register at the associated bit position |
| 0 | Disable the XOR input to the shift register at the associated bit position |


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

## 29.6.18. CRC Final XOR Register

**Name:** CRCFXOR
**Offset:** 0x44
**Reset:** 0x00000000
**Property:** R/W

**Table 29-43.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRCFXOR[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCFXOR[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCFXOR[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCFXOR[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 – CRCFXOR[31:0]**

CRCSUM is compared to the XOR of CRCACC and CRCFXOR.
When CRCMODE.PLEN32=0: CRCFXOR[31:16] is don't care. CRCFXOR[15:0] is the Final XOR.
When CRCMODE.PLEN32=1: CRCFXOR[31:0] is the Final XOR.

| Value | Description |
|---|---|
| 1 | Enable the Final XOR at the associated bit position |
| 0 | Disable the Final XOR at the associated bit position |


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

### 29.6.19. CRC Checksum Register

**Name:** CRCSUM
**Offset:** 0x48
**Reset:** 0x00000000
**Property:** R/W

**Table 29-44.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CRCSUM[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCSUM[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCSUM[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 |
|---|---|---|---|---|---|---|---|---|
| | CRCSUM[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 – CRCSUM[31:0]** CRC Checksum

CRCSUM is compared to the XOR of CRCACC and CRCFXOR after CRCMLEN bytes have been computed.
When CRCMODE.PLEN32=0: CRCFXOR[31:16] is don't care. CRCFXOR[15:0] is the CRC checksum.
When CRCMODE.PLEN32=1: CRCFXOR[31:0] is the CRC checksum.


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

## 29.6.20. Flash ECC Fault Control Register

**Name:** FFLTCTRL
**Offset:** 0x4c
**Reset:** 0x00000000
**Property:** R/W

**Table 29-45.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | FLTEN | FLTRST |
| Access | | | | | | | R/W | R/S/HC |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – FLTEN** Fault Enable

When FLTEN = 1, any write to a FFLT registers labeled as Enable-Protected is discarded and an error is generated.

| Value | Name | Description |
|---|---|---|
| 1 | ENABLE | ECC Fault Injection Enabled (module performs operation selected by FFLTMODE.FLTMD) |
| 0 | DISABLE | ECC Fault Injection Disabled |

**Bit 0 – FLTRST** Fault Reset

| Value | Name | Description |
|---|---|---|
| 1 | RESET | Resets all FLT SFR bits. |
| 0 | NONE | No Action |


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

## 29.6.21. Flash ECC Fault Mode Register

**Name:** FFLTMODE
**Offset:** 0x50
**Reset:** 0x00000000
**Property:** R/W

**Table 29-46.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | FLTMD[2:0] | | | | CTLFLT[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 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bits 14:12 – FLTMD[2:0]** Fault Mode Control
**Note:** Write Protected when FLTEN = 1.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Fault Injection Disabled |
| 2 | CAPT | Fault Capture Mode Enabled - Captures the address (in FLTADR) and Syndrome in (FLTSYN) |
| 4 | SINGLEREAD | Single Fault Injection (at bit selected by FLT1PTR) for Reads |
| 5 | DOUBLEREAD | Double Fault Injection for Reads |
| 6 | SINGLEWRITE | Single Fault Injection (at bit selected by FLT1PTR) for Writes |
| 7 | DOUBLEWRITE | Double Fault Injection for Writes |

**Bits 10:8 – CTLFLT[2:0]**
If Fault Injection Mode, this defines the bit on which a fault is injected for CTL.
**Note:** Write Protected when FLTEN = 1.

If FLTMD = 1xx and FLTEN = 1:

| Value | Description |
|---|---|
| 1 | Inject a Fault on to the associated ECC/Parity Control bits (CTL[n]) |
| 0 | No Fault Injected |


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

## 29.6.22. Flash ECC Fault Pointer Register

**Name:** FFLTPTR
**Offset:** 0x54
**Reset:** 0x00000000
**Property:** R/W

**Table 29-47.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | FLTPTR2[8] |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FLTPTR2[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | FLTPTR1[8] |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FLTPTR1[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 0:8, 9:17 – FLTPTR**

If Fault Injection Mode, this is the pointer to the bit in the parity calculation vector on which to generate a fault.

**Table 29-48.**

| Value | Descripton |
|---|---|
| 00000000 | Inject fault on Bit 0 |
| 00000001 | Inject fault on Bit 1 |
| 010001000 | Inject fault on Bit 136 (max bit for 128-bit data + 9-bit Parity) |
| 010001000 | Don't Use - UNDEFINED RESULT (above max bit) |


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

## 29.6.23. Flash ECC Fault Address Register

**Name:** FFLTADR
**Offset:** 0x58
**Reset:** 0x00000000
**Property:** R/W

**Table 29-49.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FLTADR[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 |
|---|---|---|---|---|---|---|---|---|
| | FLTADR[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 |
|---|---|---|---|---|---|---|---|---|
| | FLTADR[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 27:0 – FLTADR[27:0]**

If Fault Injection Mode, this is the System Physical Address at which to inject fault(s) Note: Not all bits in this register are writable. FFLTADR[27:24] determines which flash region (CFM, BFM, PFM) is targeted and always exist. For FFLTADR[23:0] only the number of bits required to address the largest flash region (PFM) exists. Bits that don't exist are treated as R-0.
This is a byte address but aligned to 16 bytes and FLTADR[3:0] are always 0.
In Fault Injection Mode this is the System Physical Address at which to inject fault(s).

**Notes:**

1. This is a byte address but aligned to 16 bytes and FLTADR[3:0] are always 0.
2. If FLTADR is within VSS Page space, FLT logic is disabled.


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

## 29.6.24. Flash ECC Fault Capture Address Register

**Name:** FFLTCAP
**Offset:** 0x5c
**Reset:** 0x00000000
**Property:** R

**Table 29-50.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | FLTADR[27:24] | | | |
| Access | | | | | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FLTADR[23:16] | | | | | | | |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | FLTADR[15:8] | | | | | | | |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | FLTADR[7:0] | | | | | | | |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 27:0 – FLTADR[27:0]**

If Fault Capture Mode, this is the Flash Physical Address at which a fault was detected Note: Not all bits in this register exist. FFLTADR[27:24] determines which Flash region (CFM, BFM, PFM) has been captured and always exist. For FFLTADR[23:0] only the number of bits required to address the largest flash region (PFM) exists. Bits that do not exist are treated as R-0.
In Fault Capture Mode this is the Flash Physical Address at which a fault was detected.


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

## 29.6.25. Flash ECC Fault Parity Register

**Name:** FFLTPAR
**Offset:** 0x60
**Reset:** 0x00000000
**Property:** R

**Table 29-51.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DEDOUT | | | | | | | SECOUT[8] |
| Access | R/HS/HC | | | | | | | R/HS/HC |
| Reset | 0 | | | | | | | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SECOUT[7:0] | | | | | | | |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DEDIN | | | | | | | SECIN[8] |
| Access | R/HS/HC | | | | | | | R/HS/HC |
| Reset | 0 | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SECIN[7:0] | | | | | | | |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 31 – DEDOUT**

For Flash Writes this value is based on write data and the calculated SEC Parity bits. For Flash Reads this value is based on read data and the calculated SEC Parity bits.

**Notes:**

1. See the *Flash ECC Vector* table for calculation vector bit order vs data bit order vs control bit order.
2. “DED”, “Overall Parity”, and “Parity[0]” are used interchangeably.

**Bits 24:16 – SECOUT[8:0]**

For Flash Writes this value is based on write data. For Flash Reads this value is based on read data. SECOUT[8] is always 0.

**Notes:**

1. See the *Flash ECC Vector* table for calculation vector bit order vs data bit order vs control bit order.
2. SECOUT[8].
3. The terms “SEC*[8:0]” and “Parity[9:1]” are used interchangeably.


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

**Bit 15 – DEDIN**

For Flash Writes this value is always 0. For Flash Reads this value is the overall parity read from Flash. Note: See TableHardSpace2-4 for calculation vector bit order vs data bit order vs control bit order. Note: This document refers to DED, Overall Parity, and Parity[0] interchangeably.
For Writes this value is always 0.
For Reads this value is the overall parity read from flash.

**Notes:**

1. See the *Flash ECC Vector* table for calculation vector bit order vs data bit order vs control bit order.
2. "DED", "Overall Parity", and "Parity[0]" are used interchangeably.

**Bits 8:0 – SECIN[8:0]**

For Flash Writes this value is always 0. For Flash Reads this value is the Single Error Parity bits read from Flash.
SECIN[8] is always 0.

**Notes:**

1. See the *Flash ECC Vector* table for calculation vector bit order vs data bit order vs control bit order.
2. SECIN[8] is always 0.
3. The terms "SEC*[8:0]" and "Parity[9:1]" are used interchangeably.


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

## 29.6.26. Flash ECC Fault Syndrome Register

**Name:** FFLTSYN
**Offset:** 0x64
**Reset:** 0x00000000
**Property:** R

**Table 29-52.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PERR[3:0] | | | | | CTLSTAT[2:0] | | |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CERR | DSERR[1:0] | |
| Access | | | | | | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | DEDSYN | | | | | | | SECSYN[8] |
| Access | R/HS/HC | | | | | | | R/HS/HC |
| Reset | 0 | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SECSYN[7:0] | | | | | | | |
| Access | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC | R/HS/HC |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:28 – PERR[3:0]** Parity Error

**Note:** Word size is defined by the Write Word size of the Flash, either 32-bits or 64-bits.

| Value | Name | Description |
|---|---|---|
| 1 | ERROR | Parity Error on Word n |
| 0 | NONE | No Parity Error on Word n |

**Bits 26:24 – CTLSTAT[2:0]**

**Note:** Panel is always programmed with 000 for ECC and 111 for parity.

000,001,010,100 = Calculation used ECC (i.e. programming used quad write)
011,101,110,111 = Calculation used Parity (i.e. programming used single write)

| Value | Name | Description |
|---|---|---|
| 0 | USEDECC_0 | Calculation used ECC (i.e. programming used quad write) |
| 1 | USEDECC_1 | Calculation used ECC (i.e. programming used quad write) |
| 2 | USEDECC_10 | Calculation used ECC (i.e. programming used quad write) |
| 4 | USEDECC_100 | Calculation used ECC (i.e. programming used quad write) |
| 3 | USEDPARITY_11 | Calculation used Parity (i.e. programming used single write) |
| 5 | USEDPARITY_101 | Calculation used Parity (i.e. programming used single write) |
| 6 | USEDPARITY_110 | Calculation used Parity (i.e. programming used single write) |
| 7 | USEDPARITY_111 | Calculation used Parity (i.e. programming used single write) |


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

**Bit 18 – CERR** ECC Control bit Error

| Value | Name | Description |
|---|---|---|
| 1 | SINGLE | Single Control Bit Error |
| 0 | NONE | No Control bit Error (ECCSTAT either 111 or 000) |

**Bits 17:16 – DSERR[1:0]**

Evaluates as [DERR,SERR] For Reads only when ECCSTAT = ECC

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No Errors |
| 1 | SINGLE | Single Error Corrected |
| 2 | DOUBLE_10 | Double Error Detected |
| 3 | DOUBLE_11 | Double Error Detected |

**Bit 15 – DEDSYN**

This is Overall Parity Calculated from Data and all Parity bits read from Flash.

| Value | Description |
|---|---|
| 1 | Calculated Overall Parity Differs from Read Overall Parity |
| 0 | Calculated Overall Parity Concurs with Read Overall Parity |

**Bits 8:0 – SECSYN[8:0]**

For Reads only when CTLSTAT = ECC or System bits ECCCTL[1:0]=ECC
This value is the bitwise XOR of SECIN and SECOUT.
**If DEDSYN=1:**
000000000 = No Data Error, but DED bit in Error
Non-Zero = SECSYN points to the bit position in the calculation vector that was corrected
**If DEDSYN = 0:**
000000000 = No Data Error or DED bit Error
Non-Zero = Double Error Detected.
**Note:** The number of active bits is dependent on the data width of the Flash panel.


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

# 30. Event System (EVSYS)

## 30.1. Overview

The Event System (EVSYS) allows autonomous, low-latency, and configurable communication between peripherals. Several peripherals can be configured to generate and 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 many event generators and can have multiple 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.

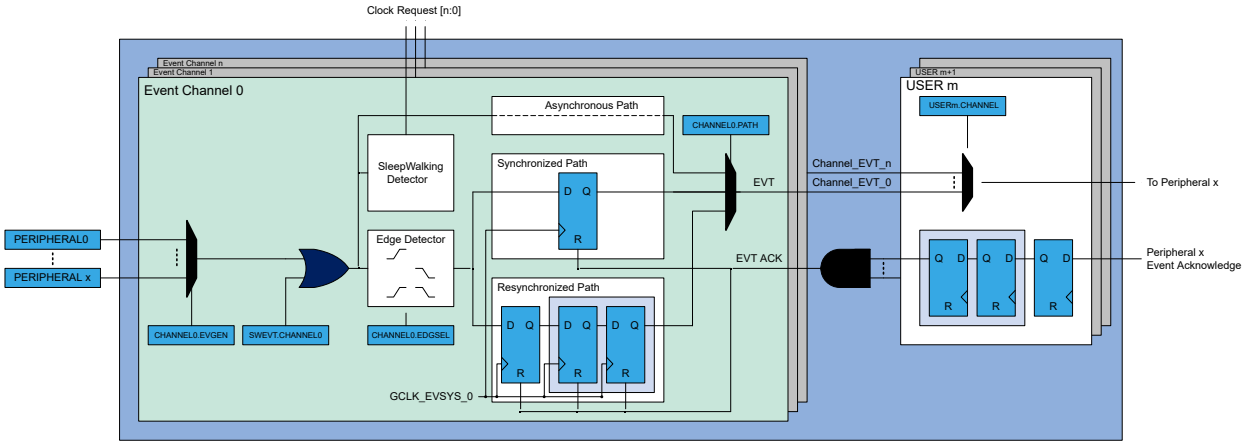
## 30.2. Features

The following are key features of the EVSYS module:

- Up to 12 configurable event channels:
  - All channels can be connected to any event generator
  - All channels provide a pure asynchronous path
  - Configurable number of channels which can provide a resynchronized path using their dedicated generic clock (GCLK_EVSYS_CHANNEL_n)
- 90 event generators
- 62 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

## 30.3. Block Diagram

**Figure 30-1.** Event System Block Diagram





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

## 30.4. Power Management

The EVSYS can be used to wake up the CPU from a sleep mode (except Backup and Off modes), even if the clock used by the EVSYS channel and the EVSYS bus clock are disabled. Refer to the *PM –Power Manager* chapter for details on the different sleep modes.

Although the clock for the EVSYS is stopped, the device can still 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 asynchronous or resynchronized.

## 30.5. Clocks

The EVSYS bus clock (CLK_EVSYS_APB) can be enabled and disabled in the Main Clock module (MCLK). The EVSYS APB BUS interface clocks, CLK_EVSYS_APB, are enabled by default on reset, see *MCLK* for additional information.

Each EVSYS channel which can be configured as 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* chapter for details.

## 30.6. Functional Description

### 30.6.1. 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 simultaneously.

A channel path can be configured in asynchronous or resynchronized mode of operation. The mode of operation must be selected based on the requirements of the application.

When using a resynchronized path, the Event System includes options to transfer events to users when rising, falling, or both edges are detected on event generators.

For additional information, refer to the *Channel Path* section of this chapter.

### 30.6.2. Basic Operation

#### 30.6.2.1. 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. The recommended sequence is as follows:

1. In the event generator peripheral, enable output of event by writing a '1' to the respective Event Output Enable bit (EO) in the peripheral's Event Control (EVCTRL) register.
2. Configure the EVSYS:
   a. Configure the Event User multiplexer by writing the respective EVSYS.USERm register, see *User Multiplexer Setup*.
   b. Configure the Event Channel by writing the respective EVSYS.CHANNELn register, see *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 (EVCTRL) register.


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

**Note:** Not all peripherals require this step.

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 (EVCTRL).

#### 30.6.2.2. 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 user multiplexers is found in the User (USERm) register description.

#### 30.6.2.3. Event System Channel

An event channel can select one event from a list of event generators. Depending on configuration, the selected event can be resynchronized or asynchronously sent to the users. When 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, for additional information, refer to the *Event System Block Diagram*

#### 30.6.2.4. Event Generators

Each event channel can receive events from all event generators. All event generators are listed in the *Peripherals Configuration Summary* section and configured 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 (that is, CHANNELn.EVGEN = 0).

#### 30.6.2.5. Channel Path

There are different ways to propagate the event from an event generator:

- Asynchronous 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 (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.

**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.


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

#### 30.6.2.6. Edge Detection

When a resynchronized path is used, edge detection must be enabled. The event system can execute edge detection in three 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).

#### 30.6.2.7. Event Latency

An event from an event generator is propagated to an event user with different latency, depending on event channel configuration.

- **Asynchronous Path:** The maximum routing latency of an external event is related to the internal signal routing and it is device dependent.
- **Resynchronized Path:** The maximum routing latency of an external event is three GCLK_EVSYS_CHANNEL_n clock cycles. Channel status and interrupt flags are set when the event routing is complete.

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.

#### 30.6.2.8. 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.
- 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 a resynchronized path. In the case of asynchronous path, the CHINTFLAGn.OVR is always read as zero.

#### 30.6.2.9. 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 resynchronized path. In the case of an asynchronous path, the CHINTFLAGn.EVD is always zero.

#### 30.6.2.10. Channel Status

The Channel Status register (CHSTATUS) shows the status of the channels when using a resynchronized path. There are two Status bits in the CHSTATUS register 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.


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

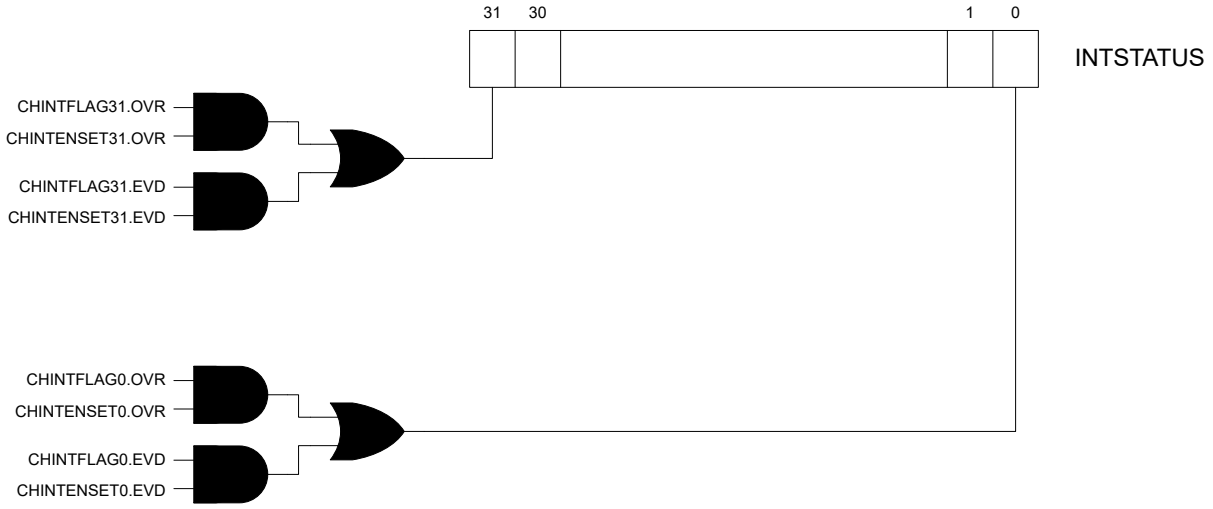
### 30.6.2.11. Software Event

A software event can be initiated on a channel by writing a '1' to the Software Event bit in the Channel register (CHANNELm.SWEVT). Then the software event can be serviced as any event generator; that is, when the bit is set to '1', an event will be generated on the respective channel.

### 30.6.2.12. Interrupt Status and Interrupts Arbitration

The Interrupt Status register stores all channels with pending interrupts, as shown below.

**Figure 30-2.** Interrupt Status Register

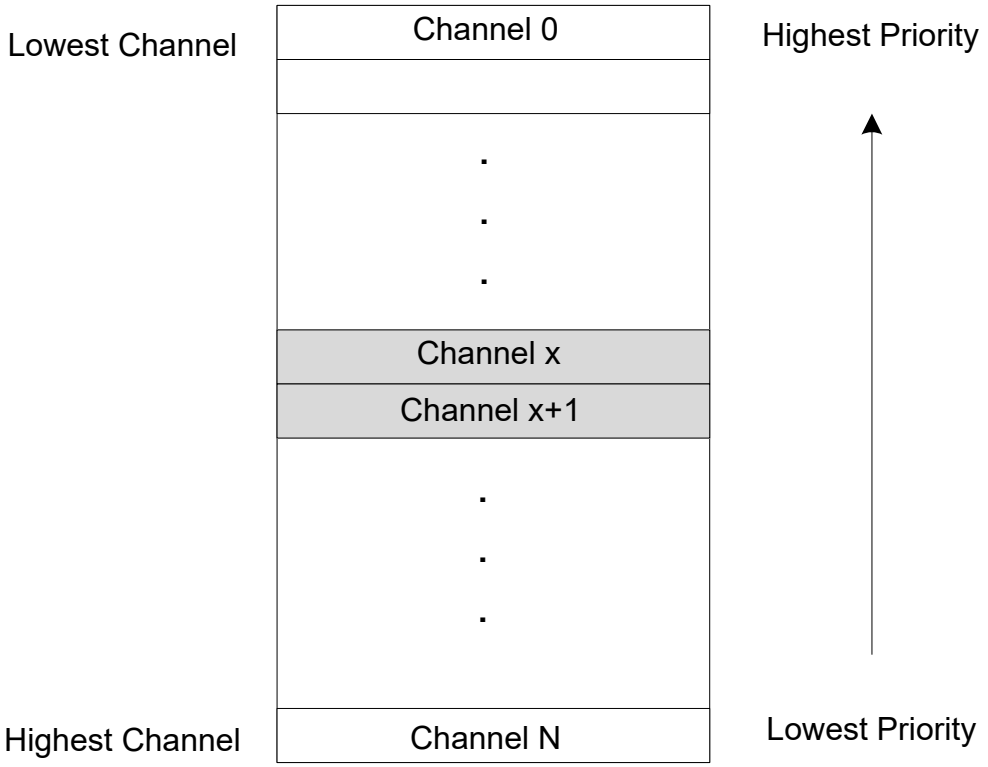



The Event System can arbitrate between all channels with pending interrupts. The arbiter can be configured to prioritize statically or dynamically by the incoming events. The priority is evaluated each time a new channel has an interrupt pending, or an interrupt has been cleared. The Channel Pending Interrupt register (INTPEND) will provide the channel number with the highest interrupt priority, and the corresponding channel interrupt flags and status bits.

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.


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

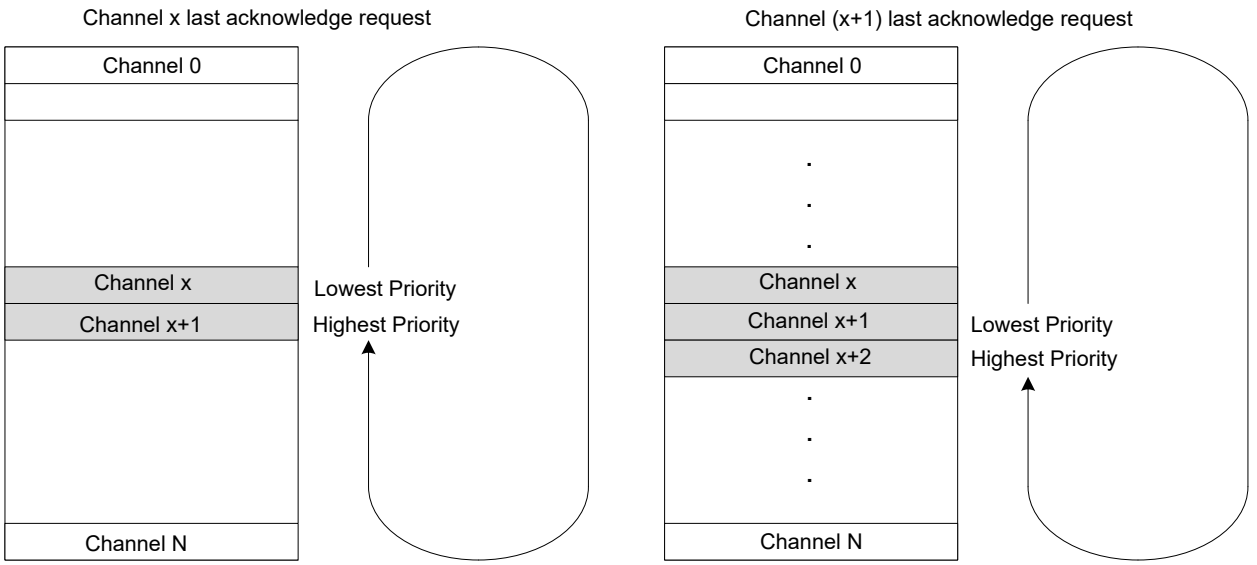
**Figure 30-3.** Static Priority




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 30-4.** 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.

### 30.6.3. Interrupts

The EVSYS has the following interrupt sources for each channel:

- Overrun Channel ‘n’ interrupt (OVR)
- Event Detected Channel ‘n’ interrupt (EVD)

These interrupt 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.

### 30.6.4. 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 the Standby bit in the Channel register (CHANNELn.RUNSTDBY) must be set to '1'. When the Generic Clock On Demand bit in the Channel


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

register (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 (that is, 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 30-1.** 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. |
| RESYNC | 0 | 1 | Run in both IDLE and STANDBY sleep modes. |
| 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. |
| 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. |

## 30.7. Peripheral Dependencies

**Table 30-2.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| EVSYS | 0x44806000 | 48 : EVD0/ OVR0<br>49 : EVD1/ OVR1<br>50 : EVD2/ OVR2<br>51 : EVD3/ OVR3<br>52 : EVD4/ OVR4<br>53 : EVD5/ OVR5<br>54 : EVD6/ OVR6<br>55 : EVD7/ OVR7<br>56 : EVD8/ OVR8<br>57 : EVD9/ OVR9<br>58 : EVD10/ OVR10<br>59 : EVD11/ OVR11 | **APB:**<br>CLKMSK[1]<br>MASK3 | **CH0:**<br>PCHCTRL[8]<br>**CH1:**<br>PCHCTRL[9]<br>**CH2:**<br>PCHCTRL[10]<br>**CH3:**<br>PCHCTRL[11]<br>**CH4:**<br>PCHCTRL[12]<br>**CH5:**<br>PCHCTRL[13]<br>**CH6:**<br>PCHCTRL[14]<br>**CH7:**<br>PCHCTRL[15]<br>**CH8:**<br>PCHCTRL[16]<br>**CH9:**<br>PCHCTRL[17]<br>**CH10:**<br>PCHCTRL[18]<br>**CH11:**<br>PCHCTRL[19] | 25<br>STATUS0<br>PERID25 | | | PD_CORE_SW |


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

## 30.8. 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 | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL |
| | | 15:8 | | | | | CHANNEL | CHANNEL | CHANNEL | CHANNEL |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 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 | CHINT | CHINT | CHINT | CHINT | CHINT | CHINT | CHINT | CHINT |
| | | 15:8 | | | | | CHINT | CHINT | CHINT | CHINT |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | BUSYCH | 7:0 | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH |
| | | 15:8 | | | | | BUSYCH | BUSYCH | BUSYCH | BUSYCH |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | READYUSR | 7:0 | READYUSR | READYUSR | READYUSR | READYUSR | READYUSR | READYUSR | READYUSR | READYUSR |
| | | 15:8 | | | | | READYUSR | READYUSR | READYUSR | READYUSR |
| | | 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 ... 0x011F | Reserved | | | | | | | | | |
| 0x0120 | USER0 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0121 ... 0x0123 | Reserved | | | | | | | | | |
| 0x0124 | USER1 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0125 ... 0x0127 | Reserved | | | | | | | | | |
| 0x0128 | USER2 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0129 ... 0x012B | Reserved | | | | | | | | | |
| 0x012C | USER3 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x012D ... 0x012F | Reserved | | | | | | | | | |
| 0x0130 | USER4 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0131 ... 0x0133 | Reserved | | | | | | | | | |
| 0x0134 | USER5 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0135 ... 0x0137 | Reserved | | | | | | | | | |
| 0x0138 | USER6 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0139 ... 0x013B | Reserved | | | | | | | | | |
| 0x013C | USER7 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x013D ... 0x013F | Reserved | | | | | | | | | |
| 0x0140 | USER8 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0141 ... 0x0143 | Reserved | | | | | | | | | |
| 0x0144 | USER9 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0145 ... 0x0147 | Reserved | | | | | | | | | |
| 0x0148 | USER10 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x0149 ... 0x014B | Reserved | | | | | | | | | |
| 0x014C | USER11 | 7:0 | | | CHANNEL[5:0] | | | | | |
| 0x014D ... 0x014F | Reserved | | | | | | | | | |
| 0x0150 | USER12 | 7:0 | | | CHANNEL[5:0] | | | | | |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0151 ... 0x0153 | Reserved | | | | | | | | | |
| 0x0154 | USER13 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0155 ... 0x0157 | Reserved | | | | | | | | | |
| 0x0158 | USER14 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0159 ... 0x015B | Reserved | | | | | | | | | |
| 0x015C | USER15 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x015D ... 0x015F | Reserved | | | | | | | | | |
| 0x0160 | USER16 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0161 ... 0x0163 | Reserved | | | | | | | | | |
| 0x0164 | USER17 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0165 ... 0x0167 | Reserved | | | | | | | | | |
| 0x0168 | USER18 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0169 ... 0x016B | Reserved | | | | | | | | | |
| 0x016C | USER19 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x016D ... 0x016F | Reserved | | | | | | | | | |
| 0x0170 | USER20 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0171 ... 0x0173 | Reserved | | | | | | | | | |
| 0x0174 | USER21 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0175 ... 0x0177 | Reserved | | | | | | | | | |
| 0x0178 | USER22 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0179 ... 0x017B | Reserved | | | | | | | | | |
| 0x017C | USER23 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x017D ... 0x017F | Reserved | | | | | | | | | |
| 0x0180 | USER24 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0181 ... 0x0183 | Reserved | | | | | | | | | |
| 0x0184 | USER25 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0185 ... 0x0187 | Reserved | | | | | | | | | |
| 0x0188 | USER26 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0189 ... 0x018B | Reserved | | | | | | | | | |
| 0x018C | USER27 | 7:0 | | | | | CHANNEL[5:0] | | | |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x018D ... 0x018F | Reserved | | | | | | | | | |
| 0x0190 | USER28 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0191 ... 0x0193 | Reserved | | | | | | | | | |
| 0x0194 | USER29 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0195 ... 0x0197 | Reserved | | | | | | | | | |
| 0x0198 | USER30 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0199 ... 0x019B | Reserved | | | | | | | | | |
| 0x019C | USER31 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x019D ... 0x019F | Reserved | | | | | | | | | |
| 0x01A0 | USER32 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01A1 ... 0x01A3 | Reserved | | | | | | | | | |
| 0x01A4 | USER33 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01A5 ... 0x01A7 | Reserved | | | | | | | | | |
| 0x01A8 | USER34 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01A9 ... 0x01AB | Reserved | | | | | | | | | |
| 0x01AC | USER35 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01AD ... 0x01AF | Reserved | | | | | | | | | |
| 0x01B0 | USER36 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01B1 ... 0x01B3 | Reserved | | | | | | | | | |
| 0x01B4 | USER37 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01B5 ... 0x01B7 | Reserved | | | | | | | | | |
| 0x01B8 | USER38 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01B9 ... 0x01BB | Reserved | | | | | | | | | |
| 0x01BC | USER39 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01BD ... 0x01BF | Reserved | | | | | | | | | |
| 0x01C0 | USER40 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01C1 ... 0x01C3 | Reserved | | | | | | | | | |
| 0x01C4 | USER41 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01C5 ... 0x01C7 | Reserved | | | | | | | | | |
| 0x01C8 | USER42 | 7:0 | | | | | CHANNEL[5:0] | | | |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01C9 ... 0x01CB | Reserved | | | | | | | | | |
| 0x01CC | USER43 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01CD ... 0x01CF | Reserved | | | | | | | | | |
| 0x01D0 | USER44 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01D1 ... 0x01D3 | Reserved | | | | | | | | | |
| 0x01D4 | USER45 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01D5 ... 0x01D7 | Reserved | | | | | | | | | |
| 0x01D8 | USER46 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01D9 ... 0x01DB | Reserved | | | | | | | | | |
| 0x01DC | USER47 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01DD ... 0x01DF | Reserved | | | | | | | | | |
| 0x01E0 | USER48 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01E1 ... 0x01E3 | Reserved | | | | | | | | | |
| 0x01E4 | USER49 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01E5 ... 0x01E7 | Reserved | | | | | | | | | |
| 0x01E8 | USER50 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01E9 ... 0x01EB | Reserved | | | | | | | | | |
| 0x01EC | USER51 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01ED ... 0x01EF | Reserved | | | | | | | | | |
| 0x01F0 | USER52 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01F1 ... 0x01F3 | Reserved | | | | | | | | | |
| 0x01F4 | USER53 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01F5 ... 0x01F7 | Reserved | | | | | | | | | |
| 0x01F8 | USER54 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01F9 ... 0x01FB | Reserved | | | | | | | | | |
| 0x01FC | USER55 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x01FD ... 0x01FF | Reserved | | | | | | | | | |
| 0x0200 | USER56 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0201 ... 0x0203 | Reserved | | | | | | | | | |
| 0x0204 | USER57 | 7:0 | | | | | CHANNEL[5:0] | | | |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0205 ... 0x0207 | Reserved | | | | | | | | | |
| 0x0208 | USER58 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0209 ... 0x020B | Reserved | | | | | | | | | |
| 0x020C | USER59 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x020D ... 0x020F | Reserved | | | | | | | | | |
| 0x0210 | USER60 | 7:0 | | | | | CHANNEL[5:0] | | | |
| 0x0211 ... 0x0213 | Reserved | | | | | | | | | |
| 0x0214 | USER61 | 7:0 | | | | | CHANNEL[5:0] | | | |


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

### 30.8.1. Control A Register

**Name:** CTRLA
**Offset:** 0x000
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 30-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 30.8.2. Software Event Register

**Name:** SWEVT
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 30-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CHANNEL | CHANNEL | CHANNEL | CHANNEL |
| Access | | | | | W | W | W | W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL | CHANNEL |
| 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 – CHANNEL** Channel x Software Selection
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.

### 30.8.3. Priority Control Register

**Name:** PRICTRL
**Offset:** 0x008
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 30-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | RREN | | | | PRI[3:0] | | | |
| Access | R/W | | | | R/W | R/W | R/W | R/W |
| 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.

### 30.8.4. Channel Pending Interrupt Register

**Name:** INTPEND
**Offset:** 0x010
**Reset:** 0x4000
**Property:** -

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.

**Table 30-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | BUSY | READY | | | | | EVD | OVR |
| Access | R | R | | | | | R/W | R/W |
| Reset | 0 | 1 | | | | | 0 | 0 |

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


© 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) 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.
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.

### 30.8.5. Interrupt Status Register

**Name:** INTSTATUS
**Offset:** 0x014
**Reset:** 0x00000000
**Property:** -

**Table 30-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CHINT | CHINT | CHINT | CHINT |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CHINT | CHINT | CHINT | CHINT | CHINT | CHINT | CHINT | CHINT |
| 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.

### 30.8.6. Busy Channel Register

**Name:** BUSYCH
**Offset:** 0x018
**Reset:** 0x00000000
**Property:** -

**Table 30-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | BUSYCH | BUSYCH | BUSYCH | BUSYCH |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH | BUSYCH |
| 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 – BUSYCH** 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.

### 30.8.7. Ready Users Register

**Name:** READYUSR
**Offset:** 0x01C
**Reset:** 0xFFFFFFFF
**Property:** -

**Table 30-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | READYUSR | READYUSR | READYUSR | READYUSR |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

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

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 – READYUSR** Ready User for Channel x
This bit is set when all event users connected to channel n are ready to handle incoming events on channel x.
This bit is cleared when at least one of the event users connected to the channel is not ready.
When the event channel x path is asynchronous, this bit is always read zero.


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

### 30.8.8. Channel n Control

**Name:** CHANNELn
**Offset:** 0x20 + n*0x08 [n=0..11]
**Reset:** 0x00008000
**Property:** RW

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.

**Table 30-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 1 | 0 | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | EVGEN[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 – 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. |


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

**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 |
|---|---|---|
| 0 | NO_EVT_OUTPUT | No event output when using the resynchronized or synchronous path |
| 1 | RISING_EDGE | Event detection only on the rising edge of the signal from the event generator when using the resynchronized or synchronous path |
| 2 | FALLING_EDGE | Event detection only on the falling edge of the signal from the event generator when using the resynchronized or synchronous path |
| 3 | BOTH_EDGES | Event detection on rising and falling edges of the signal from the event generator when using the resynchronized or synchronous path |

**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 the "USERm" register.

**Important:** When resynchronized path is enabled, event inversion feature in peripherals must not be enabled (EVCTRL.xxxINV = 0.

| Value | Name | Description |
|---|---|---|
| 0x0 | ASYNCHRONOUS | Asynchronous path |
| 0x1 | RESYNCHRONIZED | Resynchronized 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.

**Table 30-11.** Event Generators

| Value | Event Generator | Description |
|---|---|---|
| 0x01 | SUPC_LVDET | SUPC fail detection |
| 0x02 | OSCCTRL_XOSC_FAIL | XOSC fail detection |
| 0x03 | OSC32KCTRL_XOSC32K_FAIL | XOSC32K fail detection |
| 0x04 | SUPC_BOD33DET | SUPC BOD33 detection |
| 0x06-0x14 | RTC_PERx | RTC period 0-7 |
| 0x0E-0x11 | RTC_CMPx | RTC comparison x=0-3 |
| 0x12 | RTC_TAMPER | RTC tamper detection |
| 0x13 | RTC_OVF | RTC overlow |
| 0x14 | RTC_PERD | RTC |
| 0x15-0x24 | EIC_EXTINTx | EIC external interrupt, x=0-15 |
| 0x25-0x28 | DMAC_CHx | DMAC channel, x=0-3 |
| 0x29 | TC0_OVF | TC0 overflow |
| 0x2A | TC0_TRG | TC0 trigger |
| 0x2B | TC0_CNT | TC0 count |
| 0x2C-2D | TC0_MCx | TC0 match/compare, x=0-1 |
| 0x2E | TC1_OVF | TC1 overflow |
| 0x2F | TC1_TRG | TC1 trigger |
| 0x30 | TC1_CNT | TC1 count |
| 0x31-0x32 | TC1_MCx | TC1 match/compare, x=0-1 |
| 0x33 | TC2_OVF | TC2 overflow |
| 0x34 | TC2_TRG | TC2 trigger |
| 0x35 | TC2_CNT | TC2 count |
| 0x36-0x37 | TC2_MCx | TC2 match/compare |
| 0x38 | TC3_OVF | TC3 overflow |


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

**Table 30-11.** Event Generators (continued)

| Value | Event Generator | Description |
|---|---|---|
| 0x39 | TC3_TRG | TC3 trigger |
| 0x3A | TC3_CNT | TC3 count |
| 0x3B-3C | TC3_MCx | TC3 match/compare, x=0-1 |
| 0x3D | TC4_OVF | TC4 overflow |
| 0x3E | TC4_TRG | TC4 trigger |
| 0x3F | TC4_CNT | TC4 count |
| 0x40-0x41 | TC4_MCx | TC4 match/compare, x=0-1 |
| 0x42 | TC5_OVF | TC5 overflow |
| 0x43 | TC5_TRG | TC5 trigger |
| 0x44 | TC5_CNT | TC5 count |
| 0x45-0x46 | TC5_MCx | TC5 match/compare, x=0-1 |
| 0x47 | TC6_OVF | TC6 overflow |
| 0x48 | TC6_TRG | TC6 trigger |
| 0x49 | TC6_CNT | TC6 count |
| 0x4A-0x4B | TC6_MCx | TC6 match/compare, x=0-1 |
| 0x4C | ADC_CHRDY | ADC resolution ready |
| 0x4D | ADC_CMP | ADC window monitor |
| 0x4E-0x4F | AC_COMP | AC comparator |
| 0x50 | AC_WIN0 | AC window |
| 0x51 | PTC_EOC | PTC end of conversion |
| 0x52 | PTC_WCOMP | PTC window comparator |
| 0x53-0x56 | CCL0_LUTOUT | CCL0 output, x=0-3 |
| 0x57-0x5A | CCL1_LUTOUT | CCL1 output, x=0-3 |


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

### 30.8.9. Channel n Interrupt Enable Clear Register

**Name:** CHINTENCLRn
**Offset:** 0x24 + n*0x08 [n=0..11]
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 30-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EVD | OVR |
| Access | | | | | | | R/W | R/W |
| 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.

### 30.8.10. Channel n Interrupt Enable Set Register

**Name:** CHINTENSETn
**Offset:** 0x25 + n*0x08 [n=0..11]
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 30-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EVD | OVR |
| Access | | | | | | | R/W | R/W |
| 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.

### 30.8.11. Channel n Interrupt Flag Status and Clear Register

**Name:** CHINTFLAGn
**Offset:** 0x26 + n*0x08 [n=0..11]
**Reset:** 0x00
**Property:** -

**Note:** Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 30-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EVD | OVR |
| Access | | | | | | | R/W | R/W |
| 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.

### 30.8.12. Channel n Status Register

**Name:** CHSTATUSn
**Offset:** 0x27 + n*0x08 [n=0..11]
**Reset:** 0x01
**Property:** R

**Table 30-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 ‘0’.


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

### 30.8.13. User Multiplexer m Register

**Name:** USERm
**Offset:** 0x0120 + m*0x04 [m=0..61]
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 30-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

**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. Refer to the "Event User Mapping" table for finding out the m value corresponding to each module event user input.
**Notes:**

1. A value x of this bit field selects channel n = x-1.
2. By default, a channel is asynchronous. Channel resynchronous path can be enabled if resynchronous selection is written to CHANNEL.PATH bit field.

**Table 30-17.** Event User Mapping

| User Macro | User Multiplexor | USER INDEX | Description | Path Type |
|---|---|---|---|---|
| FREQM | START | 0 | Start Measurement | R |
| RTC | TAMPER | 1 | RTC Tamper | A |
| PORT | EVx | 2-5 | PORT Event x=0..3 | R |
| DMA0 | CHx-Start | 6-9 | Channel Start x=0..3 | R |
| TCC0 | EVx | 10-11 | Counter Event x=0..1 | R |
| | MCx | 12-13 | Compare/Capture x=0..1 | R |
| TCC1 | EVx | 14-15 | Counter Event x=0..1 | R |
| | MCx | 16-17 | Compare/Capture x=0..1 | R |
| TCC2 | EVx | 18-19 | Counter Event x=0..1 | R |
| | MCx | 20-21 | Compare/Capture x=0..1 | R |
| TCC3 | EVx | 22-23 | Counter Event x=0..1 | R |
| | MCx | 24-25 | Compare/Capture x=0..1 | R |
| TCC4 | EVx | 26-27 | Counter Event x=0..1 | R |
| | MCx | 28-29 | Compare/Capture x=0..1 | R |
| TCC5 | EVx | 30-31 | Counter Event x=0..1 | R |
| | MCx | 32-33 | Compare/Capture x=0..1 | R |


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

**Table 30-17.** Event User Mapping (continued)

| User Macro | User Multiplexor | USER INDEX | Description | Path Type |
|---|---|---|---|---|
| TCC6 | EVx | 34-35 | Counter Event x=0..1 | R |
| | MCx | 36-37 | Compare/Capture x=0..1 | R |
| ADC | TRIGx | 38-48 | Trigger x=0..10 | A |
| AC | SOCx | 49-50 | Start of Comparison x=0..1 | R |
| PTC | DSEQR | 51 | Restart DMA Sequencing | A |
| | STCONV | 52 | Start a Conversion | A |
| CCL0 | LUTIN_x | 53-56 | LUT Input x=0...3 | R |
| CCL1 | LUTIN_x | 57-60 | LUT Input x=0...3 | R |
| AT | TAMPSRC5 | 61 | Tamper Source | A |


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

# 31. I/O Pin Controller (PORT)

## 31.1. 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 highly configurable driver and pull settings. On the PIC32CM SG devices, if the PORT peripheral is configured as mixed-secure in the H2PB, each pin can be defined as secure or non-secured, where secured pins can only be handled by secure accesses.

All I/O pins have true read-modify-write functionality when used for general-purpose I/O; the direction or the output value of multiple 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-bit, 16-bit, or 32-bit write.

**Note:** If AVDD < 2.5v, the user must enable analog charge pumps in SUPC.VREGCTRL.CPEN.

## 31.2. Features

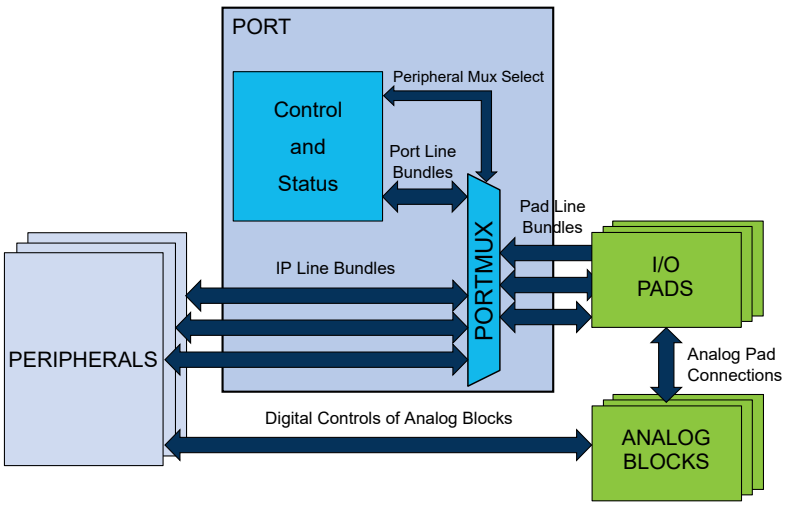
The following are key features of the PORT module:

- 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:
  - Totem-pole (push-pull)
  - Slew rate limit
  - Open drain
  - Driver Strength
- Configurable input buffer and pull settings:
  - Internal pull-up or pull-down options
  - Input sampling criteria
  - Input buffer can be disabled if not needed for lower power consumption
- 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
- Selectable secured or non-secured attribution for each individual pin (only for PIC32CM SG devices)


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

## 31.3. Block Diagram

**Figure 31-1.** PORT I/O Block Diagram




## 31.4. Signal Description

**Table 31-1.** Signal Description for PORT

| Signal name | Type | Description |
|---|---|---|
| Pxn | Digital I/O | General-purpose I/O, Port x, Pin n (Example: PA1, PC20) |

Refer to the Pinout for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

## 31.5. 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 PORTx, with letter x = A, B, C... Each pin of PORTx is identified by a number n = 0, 1, ...31 for the nth pin of that Port. In this way PORT pins are identified as 'Pxn' (for example: PA24, PC3).

Each pin may be controlled by one or more peripheral multiplexer settings, which allow 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 the Pinout for details.

On PIC32CM SG devices, each pin may be secured or nonsecured, with secured pins only accessible by secure accesses.

## 31.6. 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 of the Main Clock (MCLK).

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.

## 31.7. Power Management

During Reset, all PORT lines are configured as inputs with input buffers, output buffers and pull disabled. The PORT peripheral will continue operating in any sleep mode where its source clock is running.


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

## 31.8. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation.

## 31.9. Register Access Protection

All registers with write-access can be optionally write-protected by the Peripheral Access Controller (PAC).

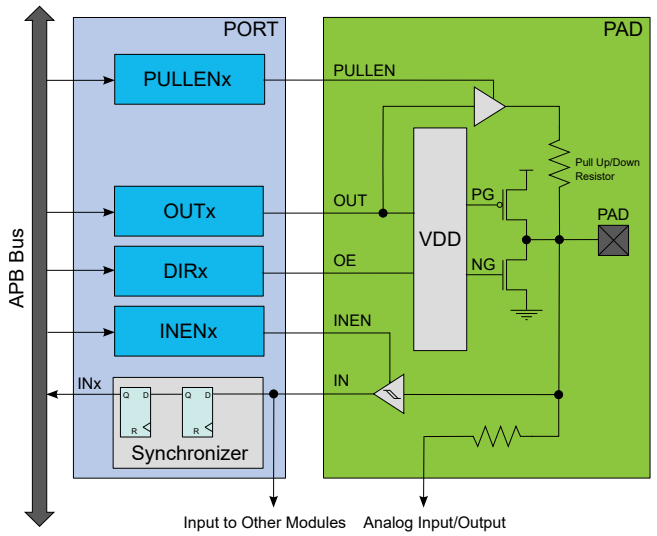
**Note:** Optional write-protection is indicated by the "PAC Write-Protection" property in the register description. Write-protection does not apply for accesses through an external debugger.

## 31.10. Analog Connections

Analog functions are connected directly between the analog blocks and the I/O pads using analog buses. However, selecting an analog peripheral function for a given pin will disable the corresponding digital features of the pad.

## 31.11. Functional Description

**Figure 31-2.** Overview of the PORT



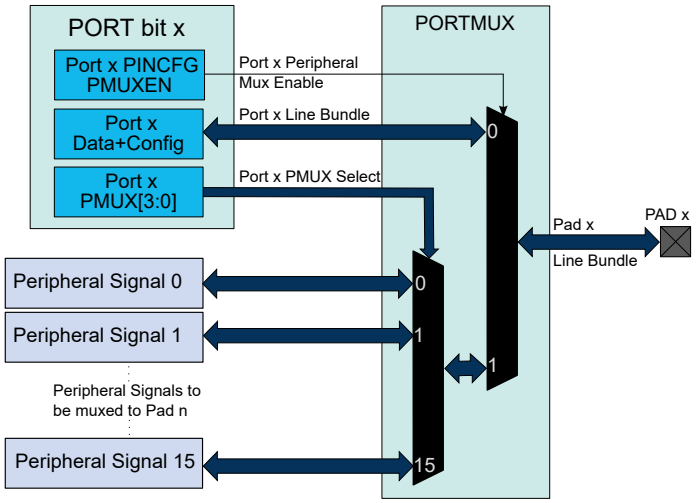

### 31.11.1. Principle of Operation

**Important:** When AVDD < 2.5v, for proper operation of the analog functions on the I/O port pins it is critical that the internal charge pumps be enabled as necessary, as defined in SUPC.VREGCTRL.CPEN.

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. The number of PORT groups may depend on the package or number of pins.


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

**Figure 31-3.** Overview of the Peripheral Functions Multiplexing




The I/O pins of the device are controlled by the 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 ‘n’ in OUT is written to '1', pin ‘n’ is driven HIGH. If bit ‘n’ in OUTx is written to '0', pin ‘n’ is driven LOW. Pin configuration can be set by the Pin Configuration (PINCFGn) registers, with n = 00, 01, ..31 representing the bit position.

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 (PINCFGn.INEN) is '0', the input value will not be sampled.

In PORT, the Peripheral Multiplexer Enable bit in the PINCFGn register (PINCFGn.PMUXEN) can be written to '1' to enable the connection between the peripheral functions and individual I/O pins. The Peripheral Multiplexing m (PMUXm, m = 0,...15) registers select the peripheral function for the corresponding pin. This will override the connection between the PORT and that I/O pin, connecting the selected peripheral signal to the particular I/O pin instead of the PORT line bundle.

On PIC32CM SG devices where the PORT peripheral is configured as mix-secure in H2PB, the security attribution of each pin in a PORT group is configured by the NONSEC register. If a bit in the NONSEC register is set to '0', the corresponding pin is configured as a secured pin and can only be handled by secure accesses. If a bit in the NONSEC register is set to '1', the corresponding pin is configured as a non-secured pin. Only secure accesses are allowed to write to the NONSEC register.

## 31.11.2. Basic Operation

### 31.11.2.1. Initialization

After reset, all standard function device I/O pads are connected to the PORT with outputs tri-stated and input buffers disabled (Hi-Z tristate condition), 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.

### 31.11.2.2. Operation

Each I/O pin, Pxn, 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 corresponds to group index 1, and so on). Within that set of registers, the pin index is 'n', from 0 to 31.

Refer to the *Pinout* for additional information on available pin configuration and PORT groups.

#### Configuring Pins as Output

To use the Pxn pin as an *output*, write bit 'n' of the DIR register to '1'. This can also be done by writing the bit 'n; in the DIRSET register to '1', this will avoid disturbing the configuration of other pins in that group. The 'n' bit in the OUT register must be written to the desired output value, this must be done first before writing the DIR bit to avoid glitching the output pin prior to setting the user desired OUT value, so the previous value of the OUT bit is not driven on the pin.

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 the Pxn pin as an *input*, the bit 'n' in the DIR register must be written to '0'. This can also be done by writing the bit 'n' 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 the bit 'n' in the IN register as soon as the INEN bit in the Pin Configuration register (PINCFGn.INEN) is written to '1'.

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 the Pxn pin as one of the available peripheral functions, the corresponding PMUXEN bit of the PINCFGn register must be '1'. The PINCFGn register for the Pxn pin is at byte offset (PINCFG0 + n).

The peripheral function can be selected by setting the PMUXO or PMUXE in the PMUXm register. The PMUXO or PMUXE is at byte offset PMUX0 + (n/2). The chosen peripheral must also be configured and enabled.

### 31.11.3. I/O Pin Configurations

The Pin Configuration register (PINCFGn) is used for additional I/O pin configuration. A pin can be set in a totem-pole, open-drain, 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 the following table:

**Table 31-2.** 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 |
| 0 | 1 | 1 | 0 | Input with pull-down |


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

**Table 31-2.** Pin Configurations Summary (continued)

| DIR | INEN | PULLEN | OUT | Configuration |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | Input with pull-up |
| 1 | 0 | X | X | Output; input disabled |
| 1 | 1 | X | X | Output; input enabled |

**Note:** X = Don't care.

### 31.11.3.1. Input Configuration

**Figure 31-4.** I/O Configuration - Standard Input

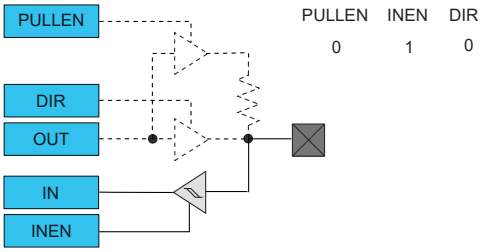
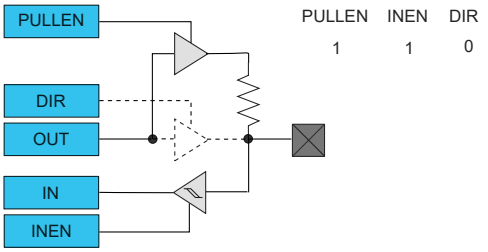



**Figure 31-5.** I/O Configuration - Input with Pull



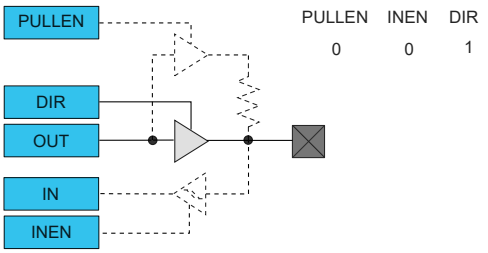

**Note:** When pull is enabled, the pull value is defined by the OUT value.

### 31.11.3.2. 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 31-6.** I/O Configuration - Totem-Pole Output with Disabled Input





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

**Figure 31-7.** I/O Configuration - Totem-Pole Output with Enabled Input

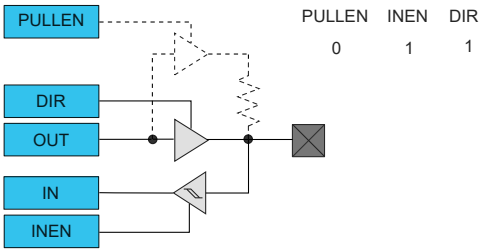
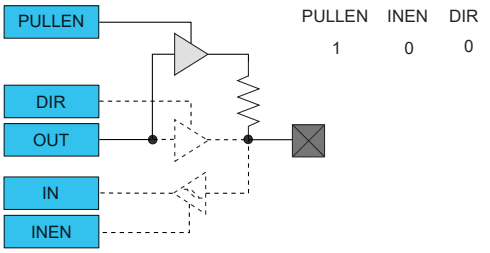


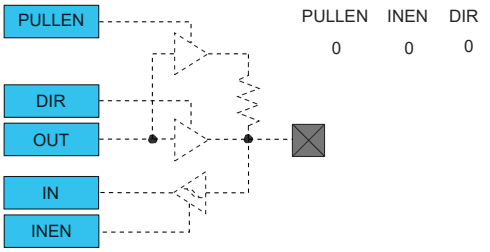

**Figure 31-8.** I/O Configuration - Output with Pull




### 31.11.3.3. Digital Functionality Disabled

Neither Input nor Output functionality are enabled.

**Figure 31-9.** I/O Configuration - Reset or Analog I/O: Digital Output, Input and Pull Disabled




### 31.11.3.4. Secure Access Rights (Only on Devices with Security Attribution)

Non-secure write to the CTRL, EVCTRL, or NONSEC register is prohibited.

Non-secure read to the CTRL or EVCTRL register will return zero with no error resulting.

Non-secure write to a bit of the DIR, DIRCLR, DIRSET, DIRTGL, OUT, OUTCLR, OUTSET, or OUTTGL register is prohibited if the corresponding bit in NONSEC is zero.

Non-secure write to a PINCFGn register or to a PMUXn register field, either directly or through a write to the WRCONFIG register, is prohibited if the corresponding bit in NONSEC is zero.

The DIR, DIRCLR, DIRSET, DIRTGL, IN, OUT, OUTCLR, OUTSET, or OUTTGL bit, PINCFGn registers, or PMUXn register fields relating to secure I/O pins (that is, the corresponding bits in NONSEC are zero), read as zero in non-secure mode, with no error resulting.

INTFLAG.NSCHK is set to 1 when the NSCHK and NONSEC registers values are different. Writing a 1 to INTFLAG.NSCHK will clear it and clear the PORT interrupt if enabled.

Secure code should initially write a 1 for all non-secure pins into both NONSEC and NSCHK registers. Then, whenever secure code writes a different value into NONSEC, INTFLAG.NSCHK will be set to 1 and a PORT interrupt will occur, if enabled. The non-secure code can then compare the values of the NONSEC and NSCHK registers to determine which PORT pins have just changed to secure or to non-secure. It should then copy the current NONSEC register value into NSCHK.


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

### 31.11.3.5. Events

The PORT allows input events to control individual I/O pins. These input events are generated by the EVSYS module and can originate from a different clock domain than the PORT module.

The PORT can perform the following actions:

- Output (OUT): The 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): The I/O pin will be set when an incoming event is detected.
- Clear (CLR): The I/O pin will be cleared when an incoming event is detected.
- Toggle (TGL): The I/O pin will toggle when an incoming event is detected.

The event is output to 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.

The event actions can be configured with the Event Action m bit group in the Event Input Control register( EVCTRL.EVACTm). Writing a '1' to a PORT Event Enable Input m of the Event Control register (EVCTRL.PORTEIm) 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 *Event System (EVSYS)*, for details on configuring the Event System.

Each event input can address 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.

Note that this truth table can be applied to any SET/CLR/TGL configuration from two to four active input events.

**Table 31-3.** 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 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.


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

## 31.12. Peripheral Dependencies

**Table 31-4.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| PORT | 0x44800000 | 41 : NSCHK0/ NSCHK1/ NSCHK2/ NSCHK3 | **APB:** CLKMSK[0] MASK27 | --- | 20 STATUS0 PERID20 | | Users: **EV0:** USER[2],Resynch **EV1:** USER[3],Resynch **EV2:** USER[4],Resynch **EV3:** USER[5],Resynch | PD_CORE_SW |


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

## 31.13. Register Summary

| 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 | | | SLEWLIM[1:0] | | ODRAIN | 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 | SLEWLIM[1:0] | | ODRAIN | PULLEN | INEN | PMUXEN |
| ... | | | | | | | | | | |
| 0x5F | PINCFG31 | 7:0 | | DRVSTR | SLEWLIM[1:0] | | ODRAIN | PULLEN | INEN | PMUXEN |


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

### 31.13.1. 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.

**Table 31-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DIR[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 |
|---|---|---|---|---|---|---|---|---|
| | DIR[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 |
|---|---|---|---|---|---|---|---|---|
| | DIR[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 |
|---|---|---|---|---|---|---|---|---|
| | DIR[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 – 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.

### 31.13.2. 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.

**Table 31-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 31.13.3. 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.

**Table 31-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 31.13.4. 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.

**Table 31-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 31.13.5. 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.

**Table 31-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 31.13.6. 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.

**Table 31-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 31.13.7. Data Output Value Set

**Name:** OUTSET
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Mix-Secure

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.

**Table 31-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[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 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[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 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[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 |
|---|---|---|---|---|---|---|---|---|
| | OUTSET[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 – 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.

### 31.13.8. 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.

**Table 31-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[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 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[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 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[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 |
|---|---|---|---|---|---|---|---|---|
| | OUTTGL[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 – 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.

### 31.13.9. Data Input Value

**Name:** IN
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** -

**Table 31-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 31.13.10. Control

**Name:** CTRL
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 31-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[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 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[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 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[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 |
|---|---|---|---|---|---|---|---|---|
| | SAMPLING[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 – 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.

### 31.13.11. Write Configuration

**Name:** WRCONFIG
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** PAC Write-Protection

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.

**Table 31-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | SLEWLIM[1:0] | | ODRAIN | PULLEN | INEN | PMUXEN |
| Access | | | W | W | W | W | W | W |
| Reset | | | 0 | 0 | 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.
Writing '1' to this bit updates the configuration of the selected pins with the written WRCONFIG.DRVST, WRCONFIG.SLEWLIM,WRCONFIG.ODRAIN, WRCONFIG.PULLEN, WRCONFIG.INEN, WRCONFIG.PMUXEN, and WRCONFIG.PINMASK values.


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

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.

**Bits 21:20 – SLEWLIM[1:0]** Output Driver Slew Rate Selection

This bit determines the new value written to PINCFGy.SLEWLIM 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 19 – ODRAIN** Open Drain Output

This bit determines the new value written to PINCFGy.ODRAIN 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.


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

| 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.

### 31.13.12. Event Input Control

**Name:** EVCTRL
**Offset:** 0x2C
**Reset:** 0x00000000
**Property:** PAC Write-Protection

There are up to four input event pins for each PORT group. Each byte of this register addresses one Event input pin.

**Table 31-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PORTEI3 | EVACT3[1:0] | | PID3[4:0] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | PORTEI2 | EVACT2[1:0] | | PID2[4:0] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | PORTEI1 | EVACT1[1:0] | | PID1[4:0] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | PORTEI0 | EVACT0[1:0] | | PID0[4:0] | | | | |
| 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, 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.

**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.

**Table 31-17.** PORT Event x Action ( x = [3..0] )

| Value | Name | Description |
|---|---|---|
| 0x0 | OUT | Output register of pin will be set to level of event. |
| 0x1 | SET | Set output register of pin on event. |
| 0x2 | CLR | Clear output register of pin on event. |


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

**Table 31-17.** PORT Event x Action ( x = [3..0] ) (continued)

| Value | Name | Description |
|---|---|---|
| 0x3 | TGL | Toggle output register of pin on event. |

**Table 31-18.** 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.

### 31.13.13. Peripheral Multiplexing

**Name:** PMUX
**Offset:** 0x30 + n*0x01 [n=0..15]
**Reset:** 0x00 except PMUX15 = 0x06
**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.

**Table 31-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PMUXO[3:0] | | | | PMUXE[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 |

**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 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 | - | Reserved |
| 0x6 | G | Peripheral function G selected |
| 0x7 | H | Peripheral function H selected |
| 0x8 | I | Peripheral function I selected |
| 0x9-0xF | - | Reserved |

The table above corresponds to a chip-specific configuration. For standalone verifications, consider that all choices from 0x0=A to 0xF=P are possible.

**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'.


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

Not all possible values for this selection may be valid. For more details, refer to I/O Multiplexing and Considerations.

| PMUXE[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 | - | Reserved |
| 0x6 | G | Peripheral function G selected |
| 0x7 | H | Peripheral function H selected |
| 0x8 | I | Peripheral function I selected |
| 0x9-0xF | - | Reserved |

The table above corresponds to a chip-specific configuration. For standalone verifications, consider that all choices from 0x0=A to 0xF=P are possible.


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

### 31.13.14. Pin Configuration

**Name:** PINCFG
**Offset:** 0x40 + n*0x01 [n=0..31]
**Reset:** 0x00
**Property:** PAC Write-Protection

**Important:** For Non-Secure accesses, read and write accesses (RW*) are allowed only if the security attribution for the corresponding I/O pin is set as Non-Secured in the NONSEC register.

**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.

**Table 31-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | DRVSTR | SLEWLIM[1:0] | | ODRAIN | PULLEN | INEN | PMUXEN |
| Access | | R/W | R/W | R/W | R// | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 0 | 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.

| Value | Description |
|---|---|
| 0 | Pin drive strength is set to normal drive strength. |
| 1 | Pin drive strength is set to stronger drive strength. |

**Bits 5:4 – SLEWLIM[1:0]** Output Driver Slew Rate Selection

| Value | Description |
|---|---|
| 0 | Slew rate control disabled (fast rise/fall time operation). |
| 1 | Slew rate control enabled (4x slower). |
| 2 | Slew rate control enabled (8x slower). |
| 3 | Slew rate control enabled (12x slower). |

**Bit 3 – ODRAIN** Open Drain Output

| Value | Description |
|---|---|
| 0 | The open drain output is disabled. |
| 1 | The open drain output is enabled. |


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

**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 via the Data Direction register (DIR) and output drive value via 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.

| 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.

# 32. Serial Communication Interface (SERCOM)

## 32.1. Overview

There are up to six instances of the Serial Communication interface (SERCOM) peripheral.

Each SERCOM provides hardware support for one of the $I^2C$, SPI, LIN, or USART protocols. It can operate as an $I^2C$ Host, $I^2C$ Client, SPI Host, SPI Client, LIN Host, LIN Client, or a USART. When a SERCOM is configured and enabled, the resources of that SERCOM are dedicated to the selected protocol and operational mode. It can use the internal generic clock or an external clock which allows it to operate in all sleep modes.

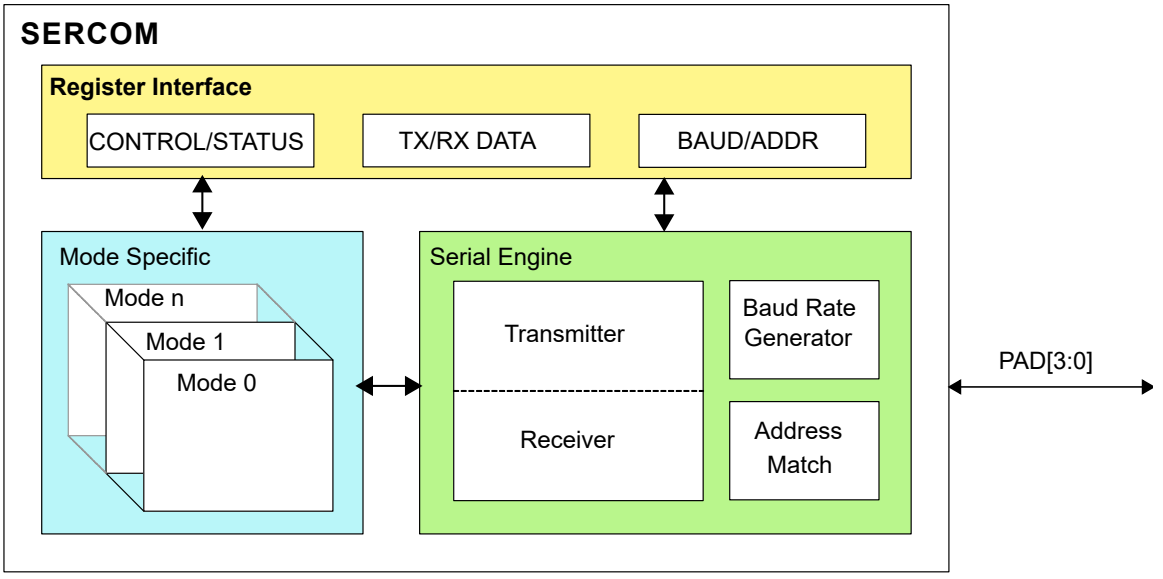
## 32.2. Features

The following are key features of the module:

- Interface for configuring operating mode:
  - Inter-Integrated Circuit ($I^2C$) two-wire serial interface
  - System Management Bus (SMBus™) compatible
  - Serial Peripheral Interface (SPI)
  - Universal Synchronous/Asynchronous Receiver/Transmitter (USART) (Plus LIN Host, LIN Client when operating in USART mode)
- Baud Rate Generator (BRG)
- Address Match/mask Logic
- Operational in all Sleep modes with an External Clock Source
- Can be used with DMA
- Up to 16-byte Internal FIFOs for Transmit and Receive

## 32.3. Block Diagram

**Figure 32-1.** SERCOM Block Diagram





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

# 32.4. Functional Description

## 32.4.1. Principle of Operation

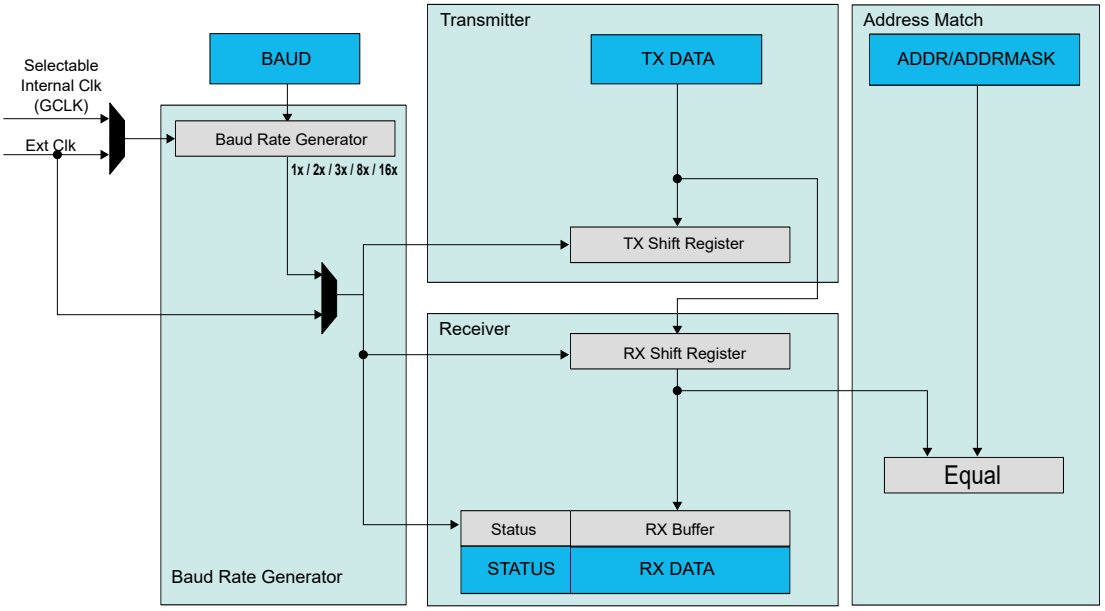
The basic structure of the SERCOM serial engine is shown in the following figure.

- The transmitter consists of a single write buffer and a Shift register.
- The receiver consists of a one-level ($I^2C$), two-level (USART, SPI) receive buffer and a Shift register.

The Baud Rate Generator (BRG) is capable of running on the GCLK_SERCOMn_CORE clock or an external clock.

Address matching logic is included for the SPI and $I^2C$ operation.

**Figure 32-2.** SERCOM Serial Engine




## 32.4.2. Basic Operation

### 32.4.2.1. Initialization

The SERCOM must be configured to the desired mode by writing the Operating Mode bits in the Control A register (CTRLA.MODE).

**Table 32-1.** SERCOM Modes

| CTRLA.MODE | Description |
|---|---|
| 0x0 | USART with external clock |
| 0x1 | USART with internal clock |
| 0x2 | SPI in client operation |
| 0x3 | SPI in host operation |
| 0x4 | $I^2C$ client operation |
| 0x5 | $I^2C$ host operation |
| 0x6-0x7 | Reserved |

### 32.4.2.2. 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.


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

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 descriptions (one for each mode of operation) for details.

- USART.CTRLA
- SPI.CTRLA
- $I^2C$ Host CTRLA
- $I^2C$ Client CTRLA

### 32.4.2.3. Clock Generation – Baud Rate Generator (BRG)

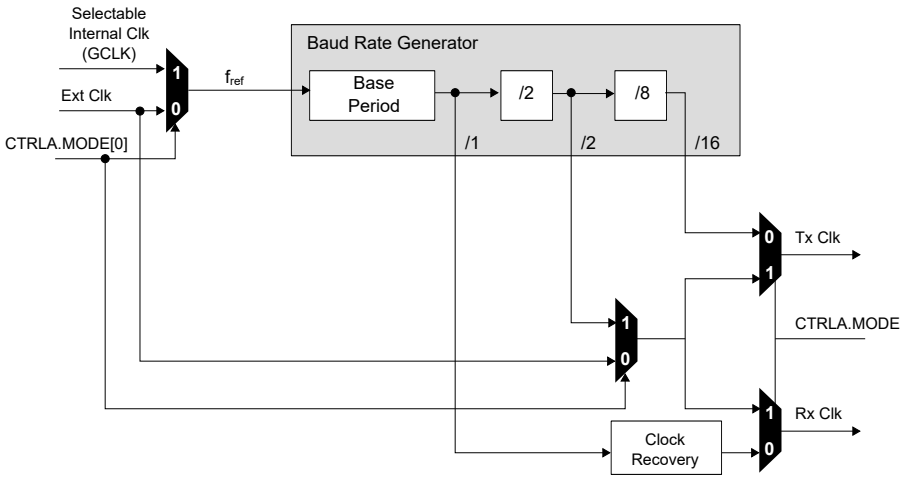
Each SERCOM contains a Baud Rate Generator as shown in the following figure, which can generate an internal clock for asynchronous or synchronous communication. The output frequency ($f_{BAUD}$) is determined by the Baud register (BAUD) setting and the baud reference clock frequency ($f_{ref}$). The 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 32-3.** Baud Rate Generator




The following table provides equations for the baud rate (in bits per second) and the BAUD register value for each operating mode.

For asynchronous arithmetic mode, the BAUD register value is 16 bits (0 to 65,535).

For asynchronous fractional mode, the BAUD register value contains 13 bits of integer (BAUD.BAUD) as well as a 3 bit fractional adjustment (BAUD.FP). In this mode the BAUD setting must be greater than or equal to 1. This mode is used for auto baud detection, For additional information, see CTRLA.FORM and CRTLA.SAMPR.

For synchronous operation, the BAUD register value is 8 bits (0 to 255). CRTLA.CMODE = 1.

**Table 32-2.** Baud Rate Equations

| Operating Mode | Condition | Baud Rate (Bits Per Second) | BAUD Register Value Calculation |
|---|---|---|---|
| Asynchronous Arithmetic | $f_{BAUD} \leq \frac{f_{ref}}{16}$ | $f_{BAUD} = \frac{f_{ref}}{16}\left(1 - \frac{BAUD}{65536}\right)$ | $BAUD = 65536 \cdot \left(1 - 16 \cdot \frac{f_{BAUD}}{f_{ref}}\right)$ |


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

**Table 32-2.** Baud Rate Equations (continued)

| Operating Mode | Condition | Baud Rate (Bits Per Second) | BAUD Register Value Calculation |
|---|---|---|---|
| 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 baud rate error is represented by the following formula:

$$\text{Error} = 1 - \left(\frac{\text{ExpectedBaudRate}}{\text{ActualBaudRate}}\right)$$

### 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* represents the data bits per frame.
- *S* represents the sum of start and first stop bits, if present.

The following table provides 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 32-3.** BAUD Register Value vs. Baud Frequency

| BAUD Register Value | Serial Engine CPF | $f_{BAUD}$ at 48 MHz Serial Engine Frequency ($f_{REF}$) |
|---|---|---|
| 0-406 | 160 | 3 MHz |
| 407-808 | 161 | 2.981 MHz |
| 809-1205 | 162 | 2.963 MHz |
| ... | ... | ... |
| 65206 | 31775 | 15.11 kHz |
| 65207 | 31871 | 15.06 kHz |
| 65208 | 31969 | 15.01 kHz |

## 32.4.3. Additional Features

### 32.4.3.1. Address Match and Mask

In either SPI client or $I^2C$ client mode, 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.

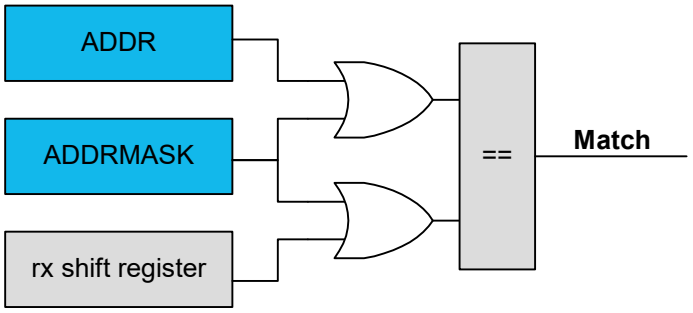
#### 32.4.3.1.1. Address With Mask (CTRLB.AMODE = 0X0)

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 if the bit values in the Shift register match the values in the Address register or the associated bit in the Address Mask register is a 1. All bits that are masked are not included in the match. Note


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

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.

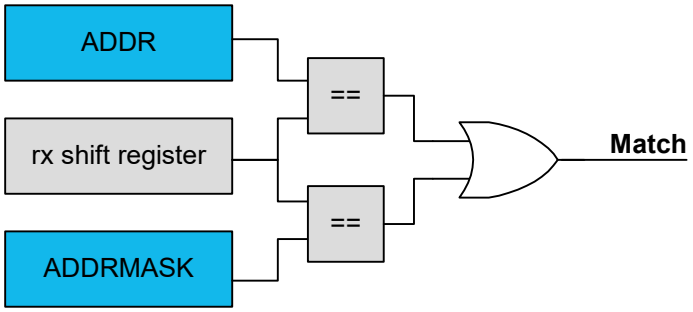
**Figure 32-4.** Address With Mask




#### 32.4.3.1.2. Two Unique Addresses (CTRLB.AMODE = 0X1)

Either of the two addresses written to ADDR and ADDRMASK will cause a match.

**Figure 32-5.** Two Unique Addresses




#### 32.4.3.1.3. Address Range (CTRLB.AMODE = 0X2)

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 32-6.** Address Range




#### 32.4.3.1.4. 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 (BRG).

The SERCOM interrupts can be used to wake-up the device from sleep modes. Refer to the different SERCOM mode chapters for additional information.

#### 32.4.3.1.5. 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.


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

# 32.5. Synchronous and Asynchronous Receiver and Transmitter (SERCOM USART)

## 32.5.1. 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, Baud Rate Generator, and either an internal or external clock.

The transmitter consists of a 16-byte FIFO, a Shift register, and control logic for different frame formats. The write buffer supports data transmission without any delay between frames. The receiver consists of a 16-byte FIFO 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.

## 32.5.2. USART Features

The USART features are as follows:

- Full-duplex Operation
- Asynchronous (with Clock Reconstruction) or Synchronous Operation
- Internal or External Clock source for Asynchronous and Synchronous Operation
- Baud Rate Generator (BRG)
- 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
- LIN Host Support
- LIN Client Support
- ISO 7816 T = 0 or T = 1 protocols for Smart Card Interfacing
- RS485 support
- Start-of-frame (SOF) detection
- Up to 16-byte Receive FIFO with FIFO enabled
- Up to 16-byte Transmit FIFO with FIFO enabled
- 32-bit Extension for Better System Bus Utilization


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

### 32.5.3. Block Diagram

**Figure 32-7.** USART Block Diagram

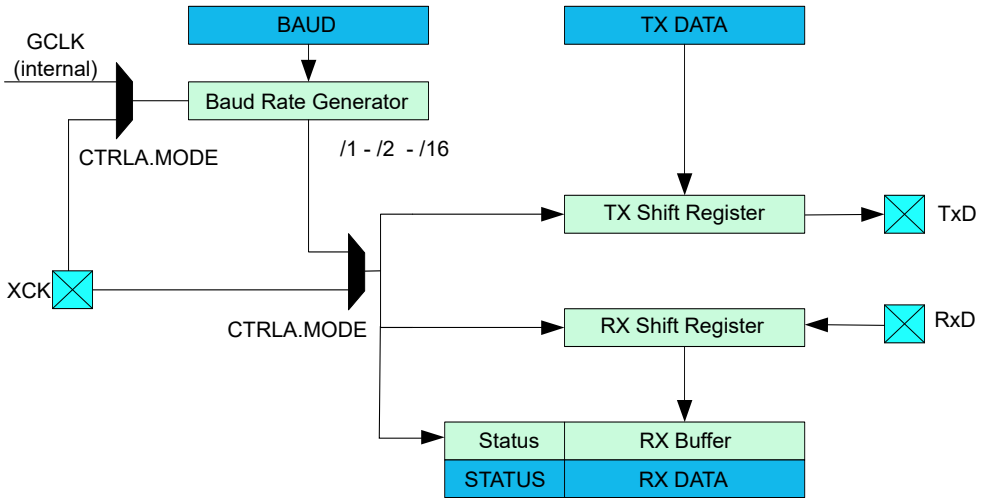



**Table 32-4.** USART Pin Configuration

| 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 pad assignments of the USART signals as shown in the table above.

### 32.5.4. Functional Description

#### 32.5.4.1. 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

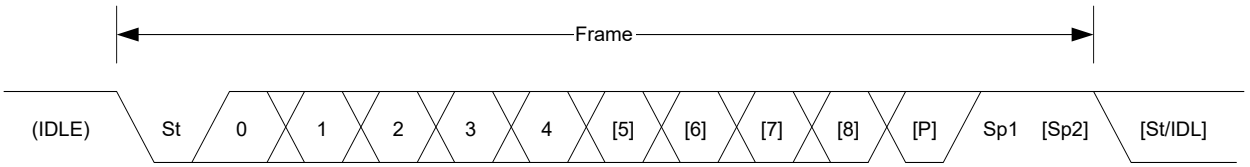
The USART data transfer is frame based. A serial frame consists of the following:

- 1 start bit
- From 5 to 9 data bits (MSB or LSB first)
- None, an even or an 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 bits of a frame, either the next frame can follow immediately, or the communication line can return to the Idle (high) state. The following figure illustrates the possible frame formats. Brackets denote optional bits.


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

**Figure 32-8.** 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.

**IDLE** – No frame is transferred on the communication line. Signal is always high in this state.

### 32.5.4.2. Basic Operation

#### 32.5.4.2.1. Initialization

The following registers are enable-protected, meaning they can only be written when the USART is disabled (CTRL.ENABLE = 0):

- The Control A register (CTRLA), except the Enable (ENABLE) and Software Reset (SWRST) bits.
- The Control B register (CTRLB), except the Receiver Enable (RXEN) and Transmitter Enable (TXEN) bits.
- The 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-first 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.


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

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).

### 32.5.4.2.2. Clock Generation and Selection

For 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.

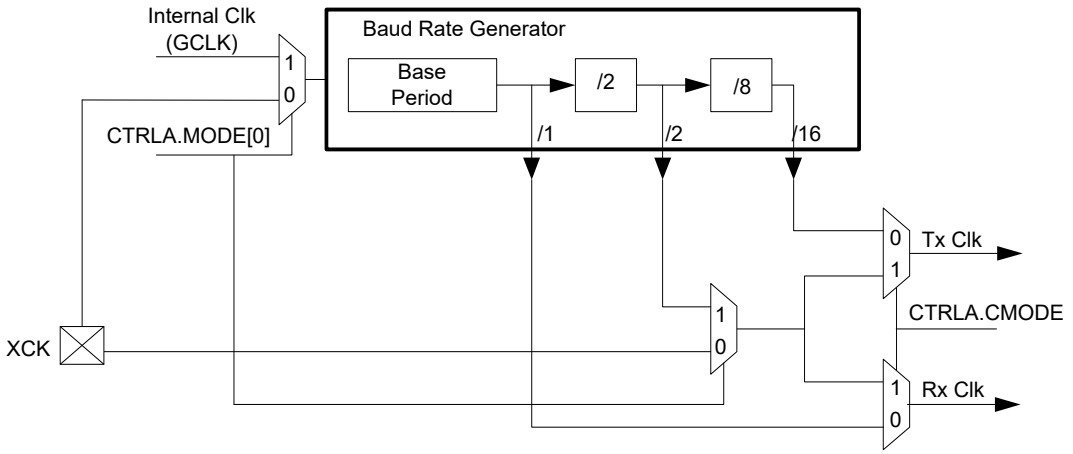
The synchronous mode is selected by writing a '1' to the Communication Mode bit in the Control A register (CTRLA.CMODE), the asynchronous mode is selected by writing a zero to CTRLA.CMODE.

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 shown 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 32-9.** Clock Generation




#### 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.

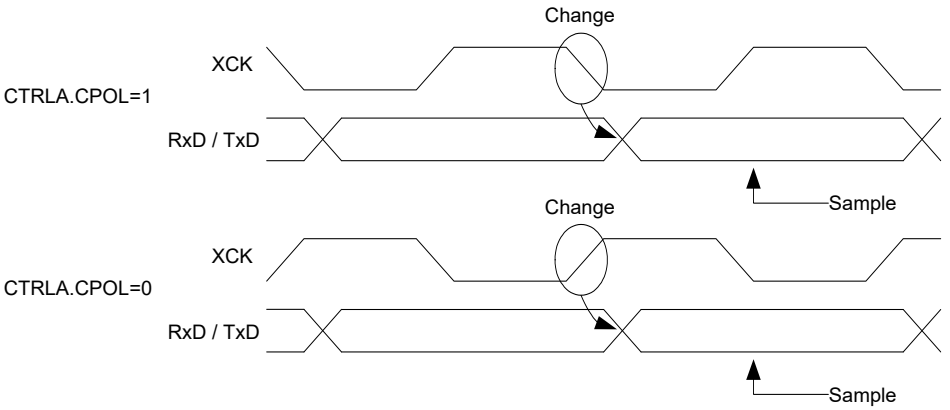
The Clock Polarity bit in the Control A register (CTRLA.CPOL) selects which XCK clock edge is used for RxD sampling, and which is used for TxD change:

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 32-10.** 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.

#### 32.5.4.2.3. 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.

#### 32.5.4.2.4. 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 bits have 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 will be generated.

The Data Register Empty flag in the Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that at least FIFO threshold (CTRLC.TXTRHOLD) locations are empty and ready for new data. The DATA register should only be written to when INTFLAG.DRE is set.

**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.

#### 32.5.4.2.5. 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 receive buffer.

The Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set when the number of bytes present in the FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD setting. An optional interrupt will 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.

### 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.

### 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. After the received data is read, STATUS.BUFOVF will be set along with INTFLAG.RXC.

### 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.

### 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, and second, 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 provided in the following table.

**Table 32-5.** 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.

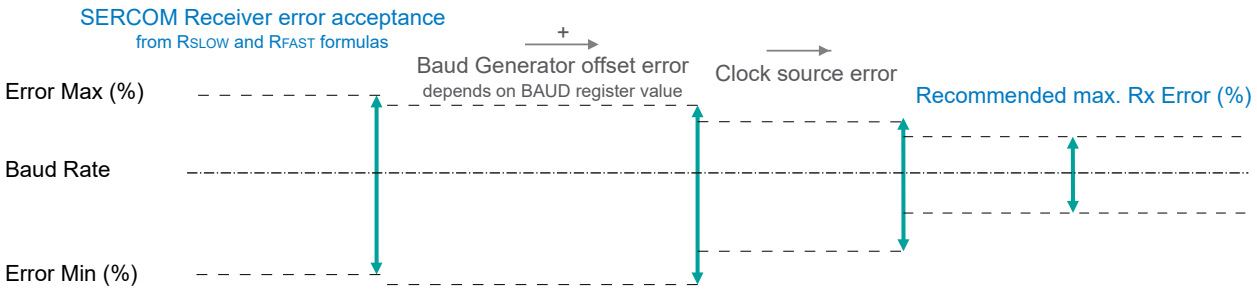
$$R_{SLOW} = \frac{(D+1)S}{S-1+D\cdot S+S_F} \quad , \quad R_{FAST} = \frac{(D+2)S}{(D+1)S+S_M}$$

Where,

- $R_{SLOW}$ 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.

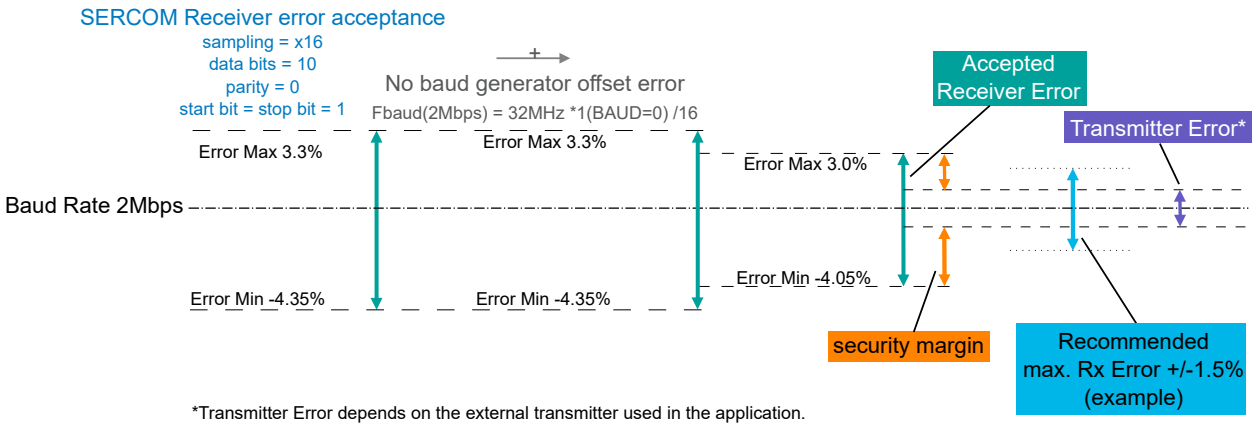
The recommended maximum Rx Error assumes that the receiver and transmitter equally divide the maximum total error and its connection to the SERCOM Receiver error acceptance is shown in this figure:

**Figure 32-11.** 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 3 Mbps:

**Figure 32-12.** USART Rx Error Calculation Example




*Transmitter Error depends on the external transmitter used in the application.
It is advised that it is within the Recommended max. Rx Error (+/-1.5% in this example).
Larger Transmitter Errors are acceptable but must lie within the Accepted Receiver Error.

### 32.5.4.3. Additional Features

#### 32.5.4.3.1. Parity

Even or odd parity can be selected for error checking by writing 0x1 or 0x5 to the Frame Format bit field in the Control A register (CTRLA.FORM).


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

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.

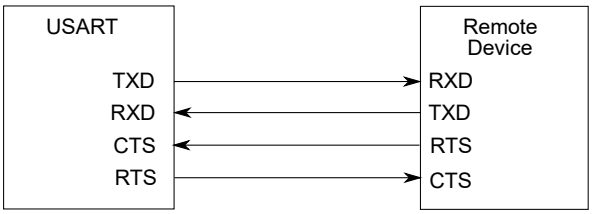
If *odd parity* is selected (CTRLB.PMODE = 1), the Parity bit of an outgoing frame is '1' if the data contains an even number of bits that are '0', making the total number of '1' odd.

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.

#### 32.5.4.3.2. 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 following figure.

**Figure 32-13.** Connection with a Remote Device for Hardware Handshaking

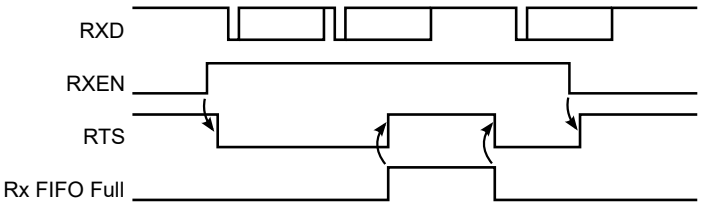



Hardware handshaking is only available in the following configuration:

- USART with internal clock (CTRLA.MODE = 1)
- Asynchronous mode (CTRLA.CMODE = 0)
- Flow control pinout (CTRLA.TXPO = 2).

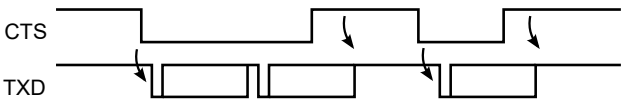
When the receiver is disabled or the receive FIFO is full, the receiver will drive the RTS pin high. This notifies the remote device to stop transfer after the ongoing transmission. Enabling and disabling the receiver by writing to CTRLB.RXEN will set/clear the RTS pin after a synchronization delay. 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 32-14.** 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 32-15.** Transmitter Behavior when Operating with Hardware Handshaking




#### 32.5.4.3.3. SERCOM USART LIN Host

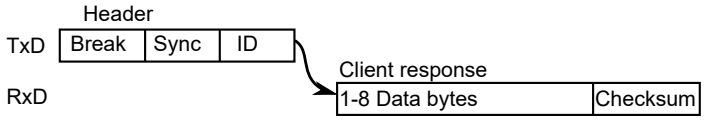
LIN host is available with the following configuration:


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

- LIN host format (CTRLA.FORM = 0x02)
- Asynchronous mode (CTRLA.CMODE = 0)
- 16x sample rate using fractional baud rate generation (CTRLA.SAMPR = 1)
- LSB is transmitted first (CTRLA.DORD = 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 32-16.** 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
- 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.
- DATA register written to 0x55. The 0x55 value (sync) is transmitted.
- DATA register written to the identifier. The identifier is transmitted.

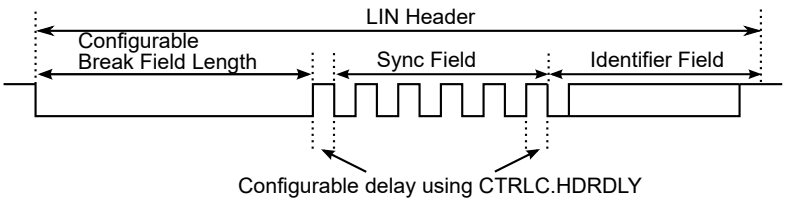
**Note:** It is recommended to use the Data Register Empty flag in the interrupt Flag Status and Clear register (INTFGLAG.DRE) to ensure no extra delay is added during the transmission of the data.

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
- 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 32-17.** LIN Header Generation




After header transmission is complete, the client responds with 1-8 data bytes plus checksum.


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

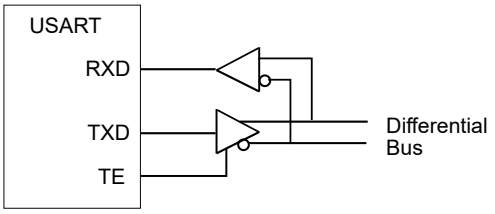
#### 32.5.4.3.4. RS485

RS485 is available with the following configuration:

- USART frame format (CTRLA.FORM = 0x00 or 0x01)
- RS485 pinout (CTRLA.TXPO = 0x3).

The RS485 feature enables control of an external line driver as shown in the following figure. While operating in RS485 mode, the transmit enable pin (TE) is driven high when the transmitter is active.

**Figure 32-18.** 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 32-19.** 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.

#### 32.5.4.3.5. 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.

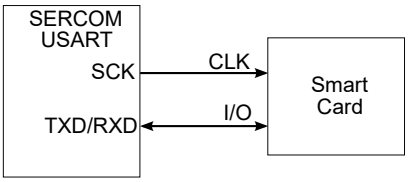
ISO 7816 is available with the following configuration:

- 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.


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

**Figure 32-20.** Connection of a Smart Card to the SERCOM USART




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 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

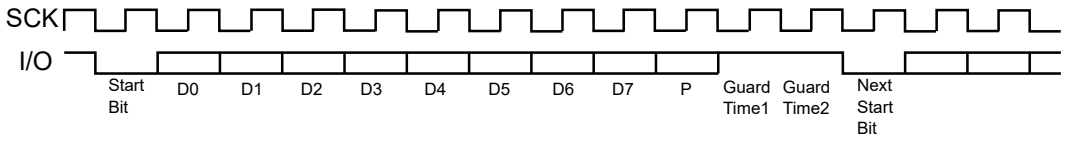
In T=0 protocol, a character is made up of:

- one start bit,
- eight data bits,
- one parity bit, and
- one guard time, which lasts two bit times

The transfer is synchronous (CTRLA.CMODE=1). 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).

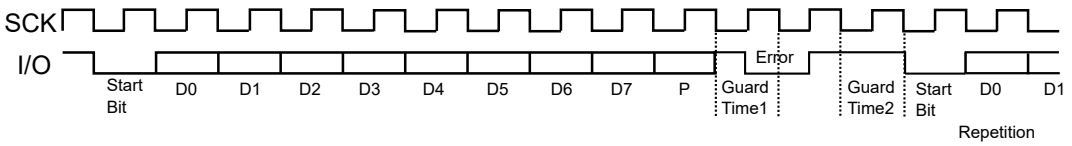
If no parity error is detected, the I/O line remains during the guard time and the transmitter can continue with the transmission of the next character, as shown in the following figure.

**Figure 32-21.** 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 following 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 32-22.** 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.


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

### 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 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

When operating in ISO7816 protocol (CTRLA.FORM=0x7) T=1 (CTRLA.FORM=0x7), the transmission is asynchronous (CTRLA.CMODE=0) with one or two stop bits. After the stop bits are sent, the transmitter does not drive the I/O line.

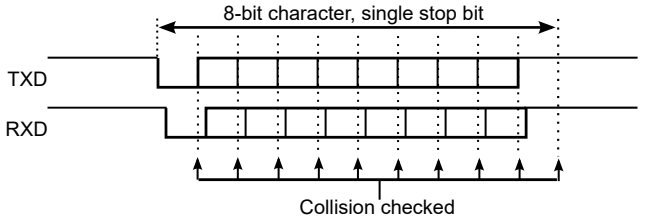
Parity is generated when transmitting and checked when receiving. Parity error detection sets the STATUS.PERR bit, and the erroneous character is written to the receive FIFO. When using T=1 protocol, the receiver does not signal errors on the I/O line and the transmitter does not retransmit.

## 32.5.4.3.6. 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.

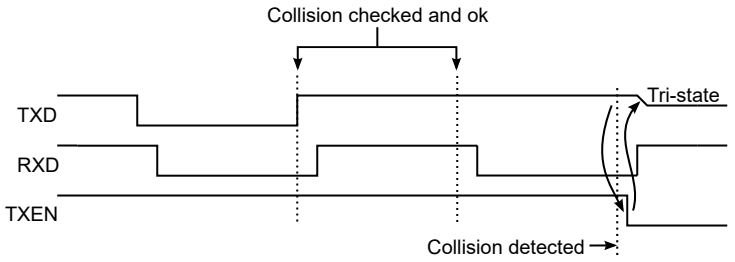
**Figure 32-23.** Collision Checking




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.


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

**Figure 32-24.** Collision Detected




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.

#### 32.5.4.3.7. 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.

#### 32.5.4.3.8. 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.

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.

#### 32.5.4.3.9. 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.


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

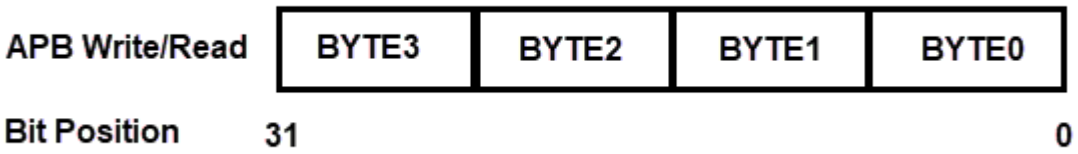
### 32.5.4.3.10. 32-bit Extension

For better system bus utilization, 32-bit data receive and transmit can be enabled separately by writing to the Data 32-bit field in the Control C register (CTRLC.DATA32B). When enabled, writes and/or reads to 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.

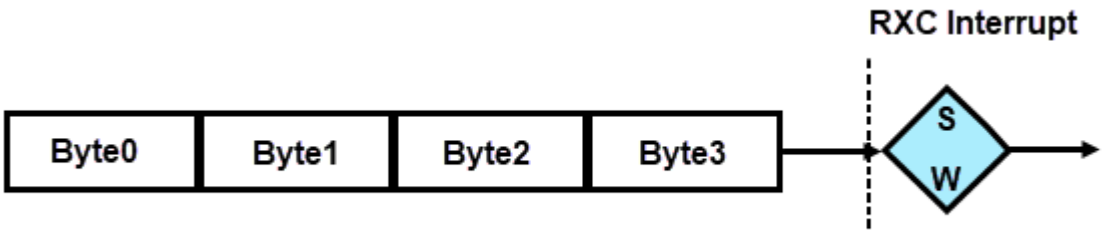
The figure below shows the order of transmit and receive when using 32-bit extension. Bytes are transmitted or received, and stored in order from 0 to 3. Only 8-bit and smaller character sizes are supported. If the character size is less than 8 bits, characters will still be 8-bit aligned within the 32-bit APB write or read. The unused bits within each byte will be zero for received data and unused for transmit data.

**Figure 32-25.** 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 32-26.** 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.


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

**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.

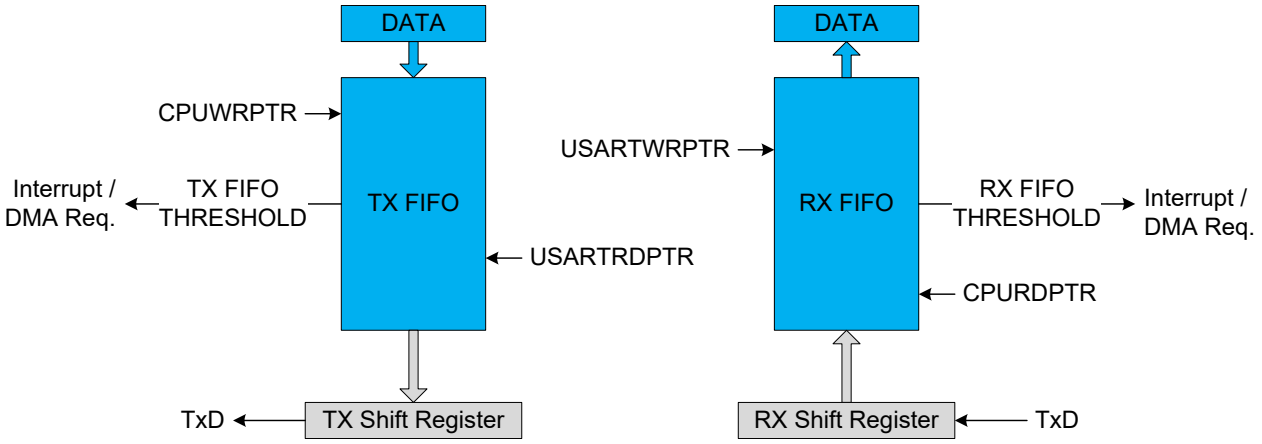
Writing the LENGTH register while a frame is in progress will produce unpredictable results. If LENGTH.LENEN is not set and a frame is not a multiple of 4 Bytes, the remainder may be lost. Attempting to use the length counter for transmit and receive at the same time will produce unpredictable results.

### 32.5.4.3.11. FIFO Operation

The USART embeds up to 16-bytes FIFO capability. The receive / transmit buffer is considered to have the FIFO mode enabled when the FIFOEN bit in CTRLC register is set to a '1' (CTRLC.FIFOEN = 1). By default, the FIFO can act as a 16-by-8-bit array, or as a 4-by-32-bit array, depending on the setting of the CTRLC.DATA32B bit.

The hardware around this array implements four pointers, called the CPU Write Pointer (CPUWRPTR), the CPU Read Pointer (CPURDPTR), the USART Write pointer (USARTWRPTR) and the USART Read pointer (USARTRDPTR). All of these pointers reset to '0'. The CPUWRPTR and CPURDPTR pointers are native to the CPU clock domain, while the USARTWRPTR and USARTRDPTR are native to the USART domain. The location pointed to by the CPUWRPTR is the current TX FIFO. The location pointed to by the CPURDPTR becomes the current RX FIFO. Writes to DATA register by the CPU will point to TX FIFO. Reads to DATA register by the CPU will point to RX FIFO. The location pointed to by the USARTWRPTR / USARTRDPTR is logically the current RX/TX shift registers.

**Figure 32-27.** FIFO Overview




The interrupts and DMA triggers are generated according to FIFO threshold settings in Control C register (CTRLC.TXTRHOLD, CTRLC.RXTRHOLD).

The Data Register Empty interrupt flag, and the DMA TX trigger respectively, are generated when the available place in the TX FIFO is equal or higher than the threshold value defined by the CTRLC.TXTRHOLD settings. The Transfer complete interrupt is generated when the TX FIFO is empty and the entire data (including the stop bits) has been transmitted.

The Receive Complete interrupt flag, and the DMA RX trigger respectively, are generated when the number of bytes present in the RX FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD settings. The ERROR interrupt flag is generated when both RX shifter and the RX FIFO are full.

The FIFO is fully accessible if the SERCOM is halted, by writing the corresponding CPU FIFO pointer in the FIFOPTR register. The RX or TX FIFO can be individually cleared, by setting the respective


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

FIFO Clear bit in the Control B register (CTRLB.FIFOCLR). The FIFO Clear must be written before data transfer begins. Writing the FIFO Clear bits while a frame is in progress will produce unpredictable results.

If the USART is halted when debugging, the RX FIFO CPU read pointer can be accessed by writing the CPURDPTR bits in FIFOPTR register (FIFOPTR.CPURDPTR). These bits will not increment if new data is read from the RX FIFO memory.

**Pointer Operation when DATA Transmission**

As in normal operation, data transmission is initiated by writing the data to be sent into the TX FIFO, by accessing the DATA register. CPUWRPTR is incremented by 1 every time the CPU writes a word to the memory array. Then, the data in TX FIFO will be moved to the shift register when the shift register is empty and ready to send a new frame, and the USARTRDPTR is incremented by 1. After the shift register is loaded with data, the data frame will be transmitted.

As long as data are present in TX FIFO (FIFOSPACE.TXSPACE != 0), a new data will be automatically loaded in the TX shift register when the previous data transmission is completed. All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then rolls over to '0'.

Depending the TX FIFO Threshold settings (CTRLC.TXTRHOLD), Interrupt Flag Status and Clear register (INTFLAG.DRE) indicates that the register is empty and ready for new data.

If the USART is halted when debugging, the CPUWRPTR pointer can be accessed by writing the CPUWRPTR bits in FIFOPTR register (FIFOPTR.CPUWRPTR). These bits will not increment if a new data is written into the TX FIFO memory.

**Pointer Operation when DATA Reception**

As in normal operation, 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 RX FIFO, and the USARTWRPTR is incremented by one. Depending the RX FIFO Threshold settings (CTRLC.RXTRHOLD), the Receive Complete interrupt flag (INTFLAG.RXC) is set, and the DATA can be read from RX FIFO. When a DATA is read, the CPURDPTR is incremented. As long as data are present in RX FIFO (FIFOSPACE.RXSPACE != 0), the CPU can read these data by accessing the DATA register. All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then rolls over to '0'.

When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The data will not be stored while BUFOVF is '1', effectively disabling the module until software reads RX FIFO.

If the USART is halted when debugging, the RX FIFO CPU read pointer can be accessed by writing the CPURDPTR bits in FIFOPTR register (FIFOPTR.CPURDPTR). These bits will not increment if a new data is read from the RX FIFO memory.

### 32.5.4.3.12. DMA and Interrupts

**Table 32-6.** Module Request for SERCOM USART

| Condition | Request | |
|---|---|---|
| | DMA | Interrupt |
| Standard (DRE): Data Register Empty<br>FIFO (DRE): at least TXTRHOLD locations in TX FIFO are empty | Yes<br>(request cleared when data is written) | Yes |
| Standard (RXC): Receive Complete<br>FIFO (RXC): at least RXTRHOLD data available in RX FIFO, or a last word available and length frame reception completed. | Yes<br>(request cleared when data is read) | Yes |
| Standard (TXC): Transmit Complete<br>FIFO (TXC): Transmit Complete and TX FIFO is empty | N/A | Yes |
| Receive Start (RXS) | N/A | Yes |


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

**Table 32-6.** Module Request for SERCOM USART (continued)

| Condition | Request | |
|---|---|---|
| | DMA | Interrupt |
| Clear to Send Input Change (CTSIC) | N/A | Yes |
| Receive Break (RXBRK) | N/A | Yes |
| Error (ERROR) | N/A | Yes |

### DMA Operation

The USART generates the following DMA requests:

- Data received (RX): The request is set when data is available in the receive FIFO or if at least RXTRHOLD data are available in the RX FIFO when FIFO operation is enabled. The request is cleared when DATA is read.
- Data transmit (TX): The request is set when the transmit buffer (TX DATA) is empty or if at least TXTRHOLD data locations are empty in the TX FIFO, when FIFO operation is enabled. The request is cleared when DATA is written.

### 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 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 the Nested Vector Interrupt Controller for details.

## 32.5.5. Sleep Mode Operation

The behavior in Sleep mode depends 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 Start and the Receive Complete interrupts can wake up the device.
- Internal clocking, CTRLA.RUNSTDBY = 0: Internal clock will be disabled, after any ongoing transfer was completed. The Receive Start and the Receive Complete interrupts can wake up the device.


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

- External clocking, CTRLA.RUNSTDBY = 0: External clock will be disconnected, after any ongoing transfer was completed. All reception will be dropped.

### 32.5.6. 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 Software Reset bit in the CTRLA register (CTRLA.SWRST)
- The Enable bit in the CTRLA register (CTRLA.ENABLE)
- The Receiver Enable bit in the CTRLB register (CTRLB.RXEN)
- The Transmitter Enable bit in the Control B register (CTRLB.TXEN)

**Note:** CTRLB.RXEN is write-synchronized somewhat differently. Refer to CTRLB in "Register Summary" section.

Required write synchronization is denoted by the *Write-Synchronized* property in the register description. If a write-synchronized register is written while a synchronization is ongoing, a Bus Error exception will be generated.

## 32.6. SERCOM Serial Peripheral Interface (SPI)

### 32.6.1. 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 the Block Diagram. SPI communications consists of a Host and at least one Client. Each side of the interface uses a separate SPI module containing a Shift register, a transmit buffer, and a receive buffer.

The SPI Host uses the SERCOM Baud Rate Generator. The SPI Client can use the SERCOM address match logic.

To better understand the documentation in this section, Labels in CAPITAL letters represent registers accessible by the CPU, while lowercase labels are internal to the SERCOM and discussed to help in explaining how the module functions.

### 32.6.2. Features

SERCOM SPI includes the following features:

- Full-duplex, four-wire interface (HICO, HOCI, 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 the HICO or HOCI pin
- Selectable LSB or MSB-first data transfer
- Can be used with DMA
- 0-bit Extension for better system bus utilization
- Framed SPI protocol support in both Host and Client operating mode, with hardware controlled FSYNC
- Internal FIFO (See Table below)
- Host operation:
  - Serial clock speed, $f_{SCK}=1/t_{SCK}$[1]
  - 8-bit clock generator
  - Hardware controlled $\overline{SS}$


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

  - 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

**Note:**

1. For $t_{SCK}$ and $t_{SSCK}$ values, refer to SPI Timing Characteristics.

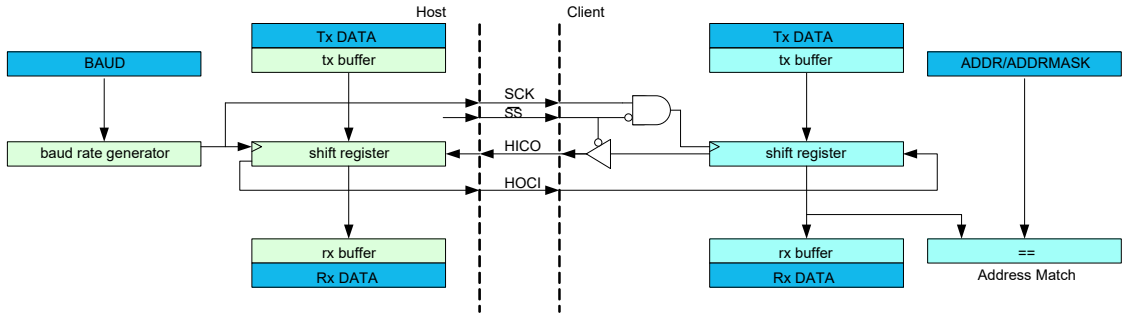
**Table 32-7.** FIFO Buffer Configuration

| FIFO Status | Transmit Buffer(s) | Receive Buffer(s) |
|---|---|---|
| Disabled or Not Implemented | Single write buffer.<br>A Shift register. | A two-level receive buffer.<br>A Shift register. |
| Enabled | 0 write FIFO buffer.<br>A Shift register. | 0 receive FIFO buffer.<br>A Shift register. |

In addition, the SPI Host uses the SERCOM baud-rate generator, while the SPI Client can use the SERCOM address match logic.

### 32.6.3. Block Diagram

**Figure 32-28.** Full-Duplex SPI Host Client Interconnection




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. Both PORT control bits PINCFGn.PULLEN and PINCFGn.DRVSTR are still effective. 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'.

In framed SPI mode (CTRLC.FRMEN = 1), the FSYNC direction is controlled by the frame type (host or client), set by FMODE bit in Control C register.

**Table 32-8.** SPI Pin Configuration

| Pin | Host SPI | Client SPI |
|---|---|---|
| HOCI | Output | Input |
| HICO | Input | Output |
| SCK | Output | Input |


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

**Table 32-8.** SPI Pin Configuration (continued)

| Pin | Host SPI | Client SPI |
|---|---|---|
| $\overline{SS}$ / FSYNC | Output (CTRLB.MSSEN=1) and (CTRLC.FRMEN = 0) | Input (CTRLC.FRMEN = 0) |
| | Output (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 0) | Output (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 0) |
| | Input (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 1) | Input (CTRLC.FRMEN = 1) and (CTRLC.FMODE = 1) |

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 in the table above.

## 32.6.4. Functional Description

### 32.6.4.1. Principle of Operation

The SPI is a high-speed synchronous data transfer interface. It allows high-speed communication between the device and peripheral devices.

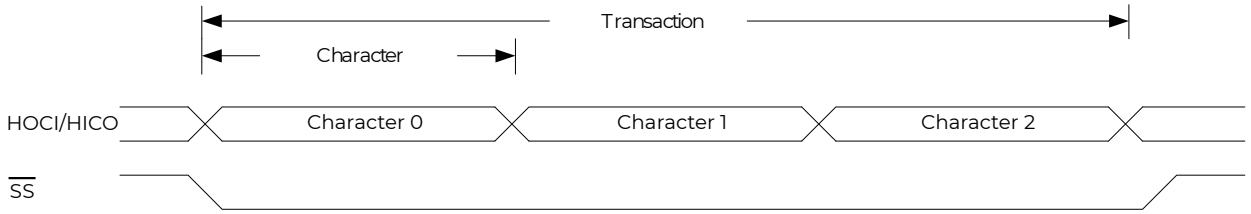
The SPI can operate as host or client. As host, the SPI initiates and controls all data transactions. The SPI is single buffered for transmitting and double buffered for receiving.

When transmitting data, the Tx buffer 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 Rx buffer, and the receiver is ready for a new character.

The SPI transaction format is shown in SPI Transaction Format diagram. Each transaction can contain one or more characters. The character size is configurable, and can be either 0 or 0 bits.

**Figure 32-29.** SPI Transaction Format




The SPI host must pull the client 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 (HOCI); data is shifted from client to host on the Host Input Client Output line (HICO).

Each time a 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.

### 32.6.4.2. Basic Operation

#### 32.6.4.2.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the SPI is disabled (CTRLA.ENABLE=0):

- Control A register (CTRLA), except Enable (CTRLA.ENABLE) and Software Reset (CTRLA.SWRST)
- Control B register (CTRLB)
- Baud register (BAUD)
- Address register (ADDR)


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

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 once the disable has finished.

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 (CTRLAS.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:
   - Select the desired baud rate by writing to the Baud register (BAUD).
   - If Hardware $\overline{SS}$ control is required, write '1' to the Host Client 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).

#### 32.6.4.2.2. 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.

#### 32.6.4.2.3. Clock and Clock Generation

**Clock**

The SERCOM bus clock (CLK_SERCOMx_APB) can be enabled and disabled in the Device 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 Device Generic Clock Controller before using the SPI.

**Note:** GCLK_SERCOMn_CORE is the only clock the user should select for the SERCOM in SPI mode.

This generic clock is asynchronous to the bus clock (CLK_SERCOMx_APB). Therefore, writing to certain registers will require synchronization between the clock domains. Refer to *Synchronization* for further details.

**Clock Generation**

In SPI host operation (CTRLA.MODE = 0x3), the serial clock (SCK) is generated internally by the SERCOM Baud Rate Generator.

In SPI mode, the Baud Rate Generator 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.


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

In SPI client operation (CTRLAS.MODE is 0x2), the clock is provided by an external host on the SCK pin. This clock is used to directly clock the SPI Shift register.

#### 32.6.4.2.4. 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 to the DATA register will update the Transmit Data register. Reading the DATA register will return the contents of the Receive Data register.

#### 32.6.4.2.5. Transfer Modes

There are four combinations of SCK phase and polarity to transfer serial data. The SPI Data Transfer modes are shown in the *SPI Transfer Modes* table, and the *SPI Transfer Modes* figure.

The 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.

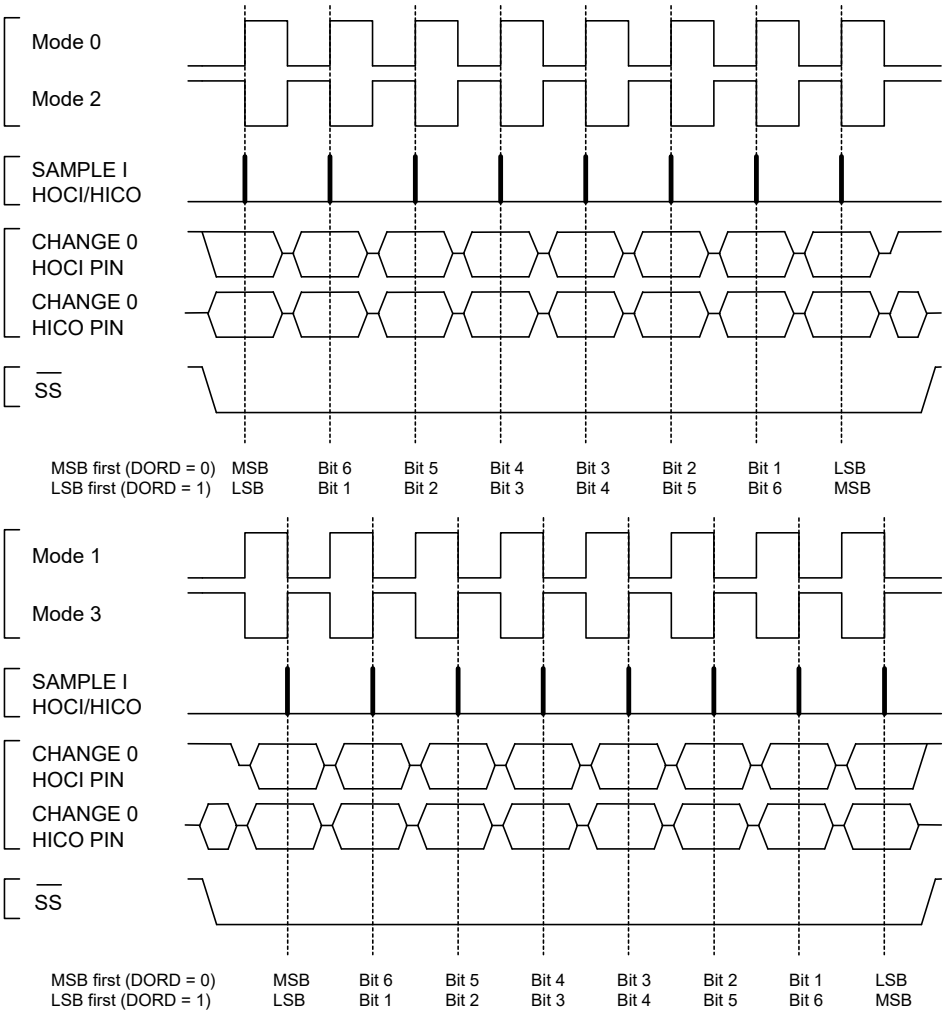
**Table 32-9.** SPI Transfer Modes

| 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 32-30.** SPI Transfer Modes




### 32.6.4.2.6. Transferring Data

#### Host Transferring Data

In Host mode (CTRLA.MODE=0x3), when Host Client Enable Select (CTRLB.MSSEN) is '1', hardware will control the $\overline{SS}$ line.

When Host Client 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.

For configurations where the FIFO is disabled or not implemented, 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. If the FIFO is enabled, INTFLAG.DRE will be set when at least FIFO threshold (CTRLC.TXTRHOLD) locations in TX FIFO are empty. Characters can be written as long as the FIFO is not full.

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 receive buffer. The transfer takes place in the same clock cycle as the last data bit is shifted in.


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

For configurations where the FIFO is disabled or not implemented, the Receive Complete Interrupt flag in the Interrupt Flag Status and Clear register (INTFLAG.RXC) will be set. For configuration where the FIFO is enabled, RXC is set when at least RXTRHOLD data are available in the RX FIFO. In both cases, the received data can be retrieved by reading DATA.

For configurations where the FIFO is disabled or not implemented, 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. For configuration where the FIFO is enabled, TXC is set when the transmit is complete and the TX FIFO is empty. When the transaction is finished, the Host must pull the $\overline{SS}$ line high to notify the Client. If Host Client Select Enable (CTRLB.MSSEN) is set to '0', the software must pull the $\overline{SS}$ line high.

**Client Transferring Data**

In Client mode (CTRLA.MODE=0x2), the SPI interface will remain inactive with the HICO line tri-stated as long as the $\overline{SS}$ pin is pulled high. Software may update the contents of Tx buffer 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.

For configurations where the FIFO is disabled or not implemented, when the content of DATA has been loaded into the Shift register, INTFLAG.DRE will be set, and new data can be written to DATA. If the FIFO is enabled, INTFLAG.DRE will be set when at least FIFO threshold (CTRLC.TXTRHOLD) locations in TX FIFO are empty. When DATA or FIFO threshold locations are empty respectively, it takes three CLK_SERCOM_ APB cycles for INTFLAG.DRE to be set.

Similar to the Host, the Client will receive one character for each character transmitted. A character will be transferred into the 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. For configurations where the FIFO is disabled or not implemented, the RXC is set when a character reception is complete. If the FIFO is enabled, the RXC is set when at least RX threshold (CTRLC.RXTRHOLD) data are received.

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 the 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.

#### 32.6.4.2.7. 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):

1. 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.
2. 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.


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

### 32.6.4.3. Additional Features

#### 32.6.4.3.1. 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 HICO 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.

If there is no match, the complete transaction is ignored.

If a 0-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.

#### 32.6.4.3.2. 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.

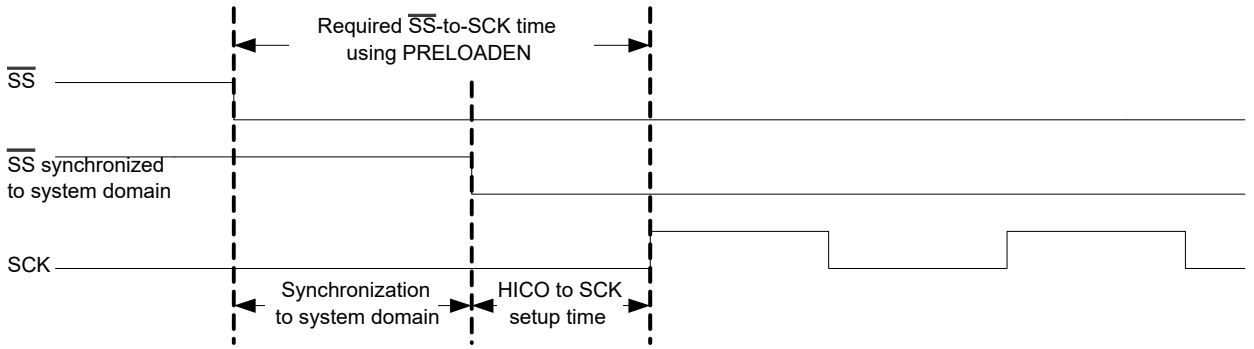
Preloading can be used to preload data into the shift register while $\overline{SS}$ is high: this eliminates sending a dummy character when starting a transaction. If the shift register is not preloaded, the current contents of the shift register will be shifted out.

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 32-31.** Timing Using Preloading

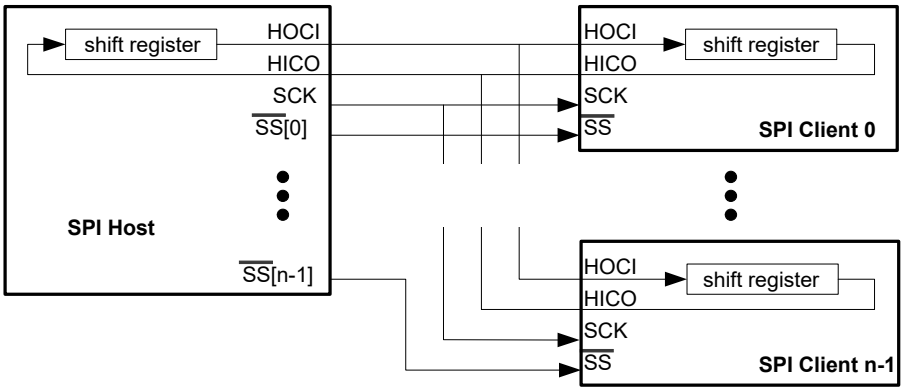



#### 32.6.4.3.3. Host with Several Clients

Host with multiple clients in parallel is only available when Host Client 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 “Multiple Clients in Parallel.” In this configuration, the single selected SPI client will drive an internal tri-stated HICO line.

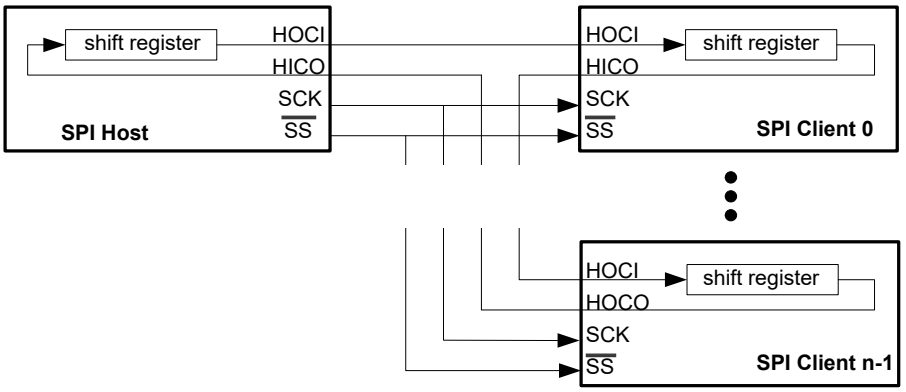

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 32-32.** Multiple Clients in Parallel




Another configuration is multiple clients in series, as in “Multiple Clients in Series.” 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 Client Select Enable bit (CTRLB.MSSEN), the $\overline{SS}$ line can be controlled either by hardware or user software and normal GPIO.

**Figure 32-33.** Multiple Clients in Series




### 32.6.4.3.4. Loopback 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.

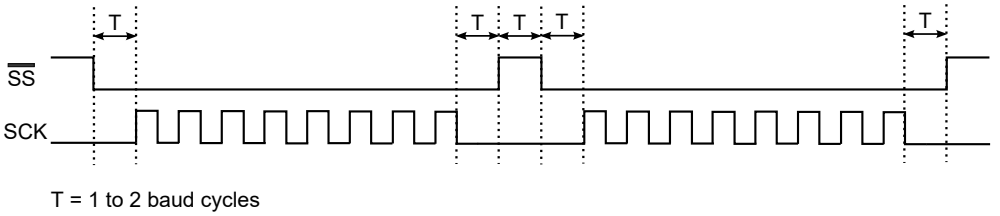
### 32.6.4.3.5. Hardware Controlled $\overline{SS}$

In Host mode, a single $\overline{SS}$ chip select can be controlled by hardware by writing the Host Client 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 32-34.** Hardware Controlled $\overline{SS}$




When CTRLB.MSSEN=0, the $\overline{SS}$ pin(s) is/are controlled by user software and normal GPIO.

#### 32.6.4.3.6. Client Select Low Detection

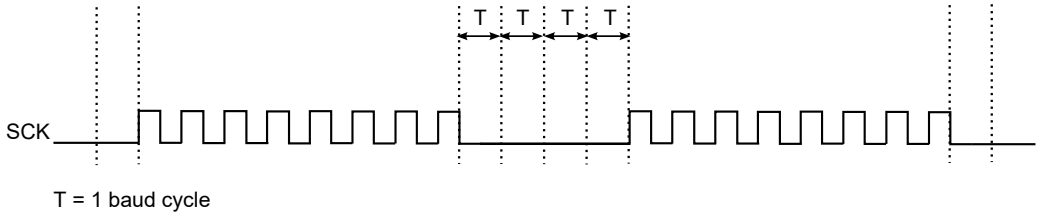
In Client mode, the SPI can wake the CPU when the client select ($\overline{SS}$) goes low. When the Client Select Low Detect is enabled (CTRLB.SSDE=1), a high-to-low transition will set the Client Select Low Interrupt flag (INTFLAG.SSL) and the device will wake-up if applicable.

#### 32.6.4.3.7. 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 following figure gives an example for CTRLC.ICSPACE=4; In this case, the SCK is inactive for 4 baud cycles.

**Figure 32-35.** Four Cycle Inter-Character Spacing Example




#### 32.6.4.3.8. 0-bit Extension

For better system bus utilization, 0-bit data receive and transmit can be enabled by writing to the Data 0-bit bit field in the Control C register (CTRLC.DATA32B=1). When enabled, write and read transaction to/from the DATA register are 0 bit in size.

If frames are not multiples of 0 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 0-bit mode. Bytes are transmitted, or received and stored, in order from 0 to 3.

Only 0-bit character size is supported.

**Figure 32-36.** 0-bit Extension Byte Ordering

| APB Write/Read | BYTE3 | BYTE2 | BYTE1 | BYTE0 |
|---|---|---|---|---|
| Bit Position | 31 | | | 0 |

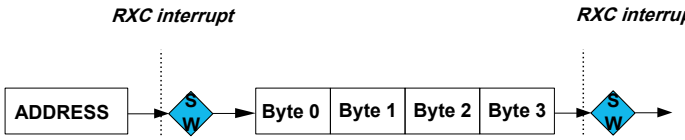
#### 0-bit Extension Client Operation

The following figure shows a transaction with 0-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


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

for every 0 Bytes transferred. For transmit, there is a 0-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 0 bits can be written to the DATA register.

**Figure 32-37.** 0-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 0 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.

#### 0-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 0 Bytes received. If LENGTH.LEN is not a multiple of 0 Bytes, the final INTFLAG.RXC is set when the final byte is received.

For transmit, there is a holding buffer for the 0-bit data in the core domain. Once DATA has been registered in the core domain, INTFLAG.DRE will be raised so that the next 0 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.

### 32.6.4.3.9. Framed SPI Operation

The SPI supports Framed SPI protocol while operating in either Host or Client mode. The CTRLC.FRMEN bit enables Framed SPI modes and causes the $\overline{SS}$ pin to be used as a frame synchronization (FSYNC) input or output pin. The state of CTRLB.MSSEN bit is then ignored.

Unlike in normal SPI mode, the serial clock is continuous (free-running) in Framed SPI mode rather than being generated only when there is data to be transmitted. The data transmission/reception starts only when the frame synchronization is generated at the FSYNC pin. The device can be either a frame host if it generates the frame sync or a frame client if it receives the frame sync at the FSYNC pin. In other words, only a frame host can generate the frame synchronization pulse.

Frame Host or Client mode is selected by clearing or setting the CTRLC.FMODE bit, respectively. The frame synchronization can be an active-high or an active-low pulse of one SCK clock duration, or frame duration, based on the CTRLC.FSLEN and CTRLC.FSPOL settings.


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

Irrespective of which device is a host and which is a client, a framed SPI data transfer begins as soon as the frame host generates the frame sync and writes the data to DATA. For full-duplex operation, the frame client should write to its buffer before the frame host does, in order to ensure that the data is ready at both ends when the data transfer begins.

### Frame Host

The frame host mode is enabled by setting CTRLC.FMODE = HOST and CTRLC.FRMEN = 1.

If the module is enabled in host operating mode (CTRLDA.MODE = 0x03), the serial clock is output at the SCK pin, regardless of whether the module is transmitting, and the FSYNC pin is driven high on the next transmit edge of the SCK clock when the Data buffer is written. Data will start transmitting on the subsequent transmit edge of the SPI clock. As long as data is available in the transmit buffer, a new frame sync is initiated after completing a transmit/receive sequence.

If the module is enabled in client operating mode (CTRLA.MODE = 0x2), the input clock at the SCK pin is continuous while the FSYNC pin and transmission are generated in the same way as in host operating mode.

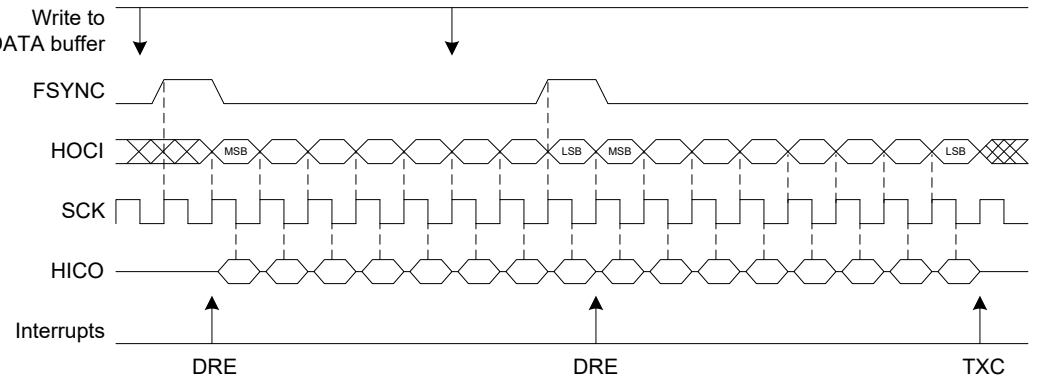
If the 0-bit extension is disabled (CTRLC.DATA32B = 0), the transmit will continue as long as data are present in the internal buffer. The Transmit Complete interrupt flag is set when there are no more data to be transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new data can be stored in the internal buffer.

If the 0-bit extension is enabled (CTRLC.DATA32B = 1), and the DATA Length Counter is disabled (LENTH.LENEN = 0), the behavior is similar to the previous mode, with the exception the DATA must be 0-bit format (word).

When the DATA Length Counter is enabled (LENTH.LENEN = 1), the length of the frame is controlled by the DATA Length field in LENGTH register (LENGTH.LEN). The Transmit Complete interrupt flag is set when there are no more data to be transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new 0-bit data can be stored in the internal buffer.

A DMA trigger is generated each time there is 0-bit internal place to store a new data. As a consequence, the DMA descriptor transfer counter must be 0-bit aligned.

**Figure 32-38.** FMODE = HOST, FSPOL = HIGH, FSLEN = STROBE, FSES = 0





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

**Figure 32-39.** FMODE = HOST, FSPOL = HIGH, FSLEN = LEVEL, FSES = 0

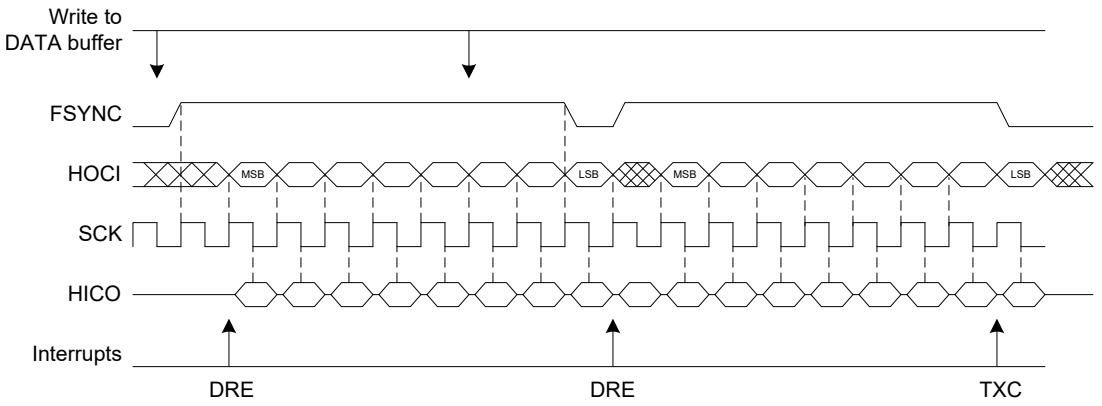



**Figure 32-40.** FMODE = HOST, FSPOL = HIGH, FSLEN = STROBE, FSES = 1

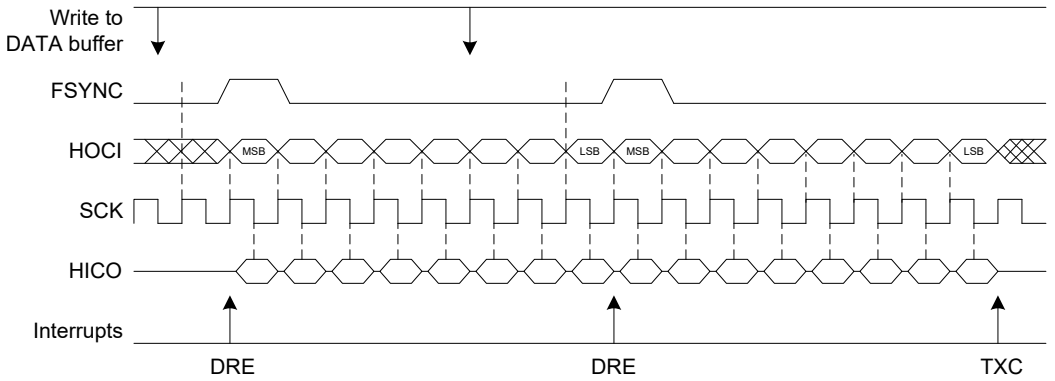
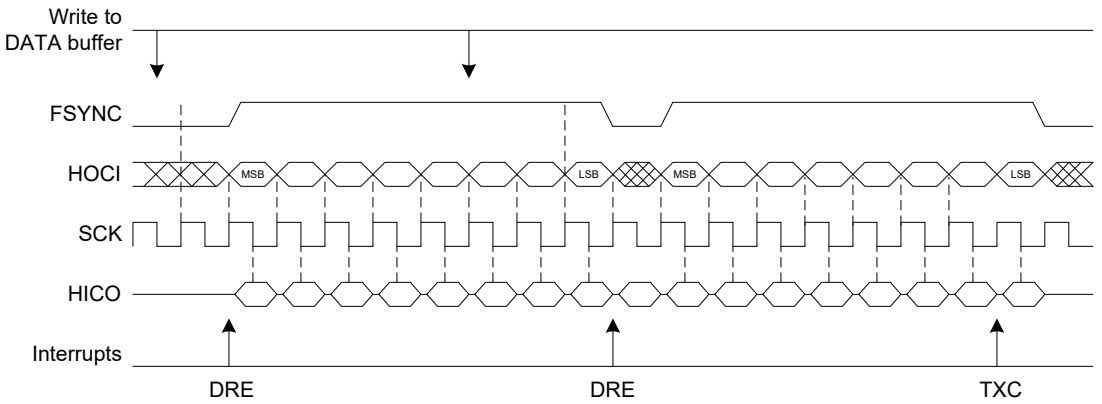



**Figure 32-41.** FMODE = HOST, FSPOL = HIGH, FSLEN = LEVEL, FSES = 1




### Frame Client

The frame client mode is enabled by setting CTRLC.FMODE = CLIENT and CTRLC.FRMEN = 1. In both host (CTRLA.MODE = 0x03) and client (CTRLA.MODE = 0x02) operating modes, the FSYNC pin is an input. The serial clock is output at the SCK pin in host operating mode, and an input in client operating mode.

When the FSYNC pin is sampled high or low (CTRLC.FSPOL = 1/0), the data is transmitted on the subsequent transmit edge of the SPI clock. The user must make sure that the correct to-be-


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

transmitted data is loaded into DATA buffer before the frame sync pulse is received at the FSYNC pin. If the FSYNC is detected and there is no data available in the buffer, the module will transmit zeroes until the end of the frame.

If the 0-bit extension is disabled (CTRLC.DATA32B = 0), one byte DATA will be sent, and a new FSYNC must be received before enabling a new transmission. The Transmit Complete interrupt flag is set when the internal shift register is empty. The Data Register Empty flag is set each time new data can be stored in the internal buffer.

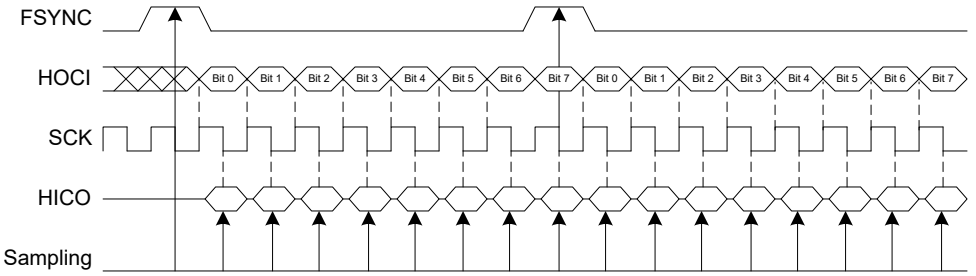
If the 0-bit extension is enabled (CTRLC.DATA32B = 1), and the DATA Length Counter is disabled (LENTH.LENEN = 0), the behavior is similar to the previous mode, with the exception the DATA will be 0-bit format (word).

When the DATA Length Counter is enabled (LENTH.LENEN = 1), the length of the frame is controlled by the DATA Length field in LENGTH register (LENGTH.LEN). The Transmit Complete interrupt flag is set when LENGTH DATA are transferred, and when the internal shift register is empty. The Data Register Empty flag is set each time a new 0-bit data can be stored in the internal buffer. A new FSYNC must be received before a new transaction starts.

If the address matching is enabled, the first data after FSYNC detection will be compared with the ADDR register. If a match occurs, the client continues to receive data. If the match fails, the data is discarded and the client waits for a new FSYNC.

A DMA trigger is generated each time there is 0-bit internal place to store a new data. As a consequence, the DMA descriptor transfer counter must be 0-bit aligned.

**Figure 32-42.** FMODE = CLIENT, FSPOL = HIGH, FSLEN = STROBE, FSES = 0




### Framed SPI Errors

If the transmit buffer is empty when the SPI must load the shift register to start (in the case of Framed Client) or continue (for either Framed Host or Client) a transfer, the SPI immediately sets the transmit underrun status bit in the STATUS (STATUS.TUR) register. If enabled, an Error interrupt flag (INTFLAG.ERROR) will also be generated.

While the SPI is in an underrun condition, the SPI transmits all zeroes until the end of the transaction (byte, word or length counter). If the SPI is a frame client, another frame sync pulse can occur before the condition clears. In this case the SPI continues to transmit zeros. If the SPI is a frame host, it then waits for software to clear the under-run condition before initiating another FSYNC regardless of the state of the DATA buffer.

When the TUR status is cleared the data present in the DATA buffer is flushed. The SPI ignores writes to the DATA after clearing the STATUS.TUR bit, until a read of STATUS register, when STATUS.TUR=0. This behavior ensures that a data service routine that is interrupted long enough to cause STATUS.TUR, cannot inadvertently start a new framed transaction after the SPI error handler has cleared the error.

For cases when software does not care or need to know about the underrun condition, CTRLC.IGNTUR = 1 provides the serial engine the ability to ignore the underrun. When an underrun occurs, the SPI still sets the corresponding STATUS and INTFLAG bits. When the SPI is either a frame client or a frame host, an underrun event still causes the SPI to transmit zeros until the end of


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

the frame. However, when the Ignore Transmit Underrun bit is set (CTRLC.IGNTUR = 1), the SPI can re-sample the underrun condition and continue to transmit data at each frame boundary. If the SPI is a frame host, new data written to the DATA buffer during a frame when an underrun condition exists does not get transmitted during that frame. But, the SPI evaluates the DATA buffer continuously after the last frame. If data is in it, the SPI generates a frame sync and transmits the data. If the SPI is a frame client, its transmit logic evaluates DATA buffer for underrun during the next FSYNC. If the buffer contains data when a valid FSYNC is detected, SPI transmits that data. If not, the SPI transmits zero data until the end of the frame.

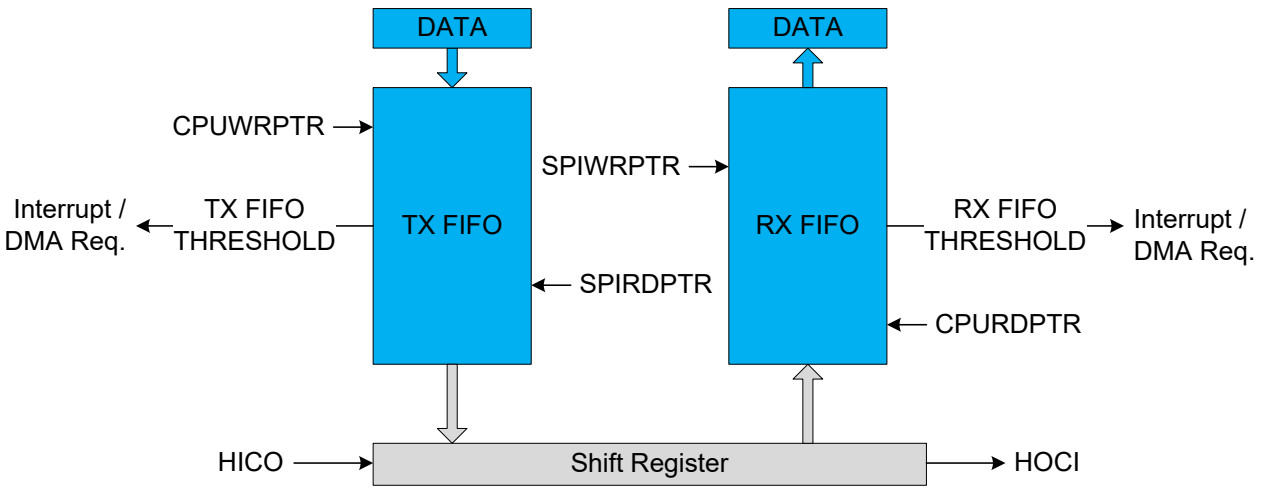
In the case of client frames, if a new FSYNC is received during an ongoing frame transaction, the transaction length error status bit is set, and the error interrupt flag is set. The module will continue to send data from the internal buffer.

### 32.6.4.3.10. FIFO Operation

The SPI embeds up to 16-bytes FIFO capability. The receive/transmit buffer is considered to have the FIFO mode enabled when the FIFOEN bit in the CTRLC register is set to a '1' (CTRLC.FIFOEN = 1). By default, the FIFO can act as a 0-bit array, or as a 0-bit array, depending on the setting of the CTRLC.DATA32B bit.

The hardware around this array implements four pointers, called the CPU Write Pointer (CPUWRPTR), the CPU Read Pointer (CPURDPTR), the SPI Write pointer (SPIWRPTR) and the SPI Read pointer (SPIRDPTR). All of these pointers reset to '0'. The CPUWRPTR and CPURDPTR pointers are native to the CPU clock domain, while the SPIWRPTR and SPIRDPTR are native to the SPI domain. The location pointed to by the CPUWRPTR is the current TX FIFO. The location pointed to by the CPURDPTR is the current RX FIFO. Writes to DATA register by the CPU will point to TX FIFO. Reads to DATA register by the CPU will point to RX FIFO. The location pointed to by the SPIWRPTR / SPIRDPTR is logically the current shift register. Physically, the receive (shift-in) portion of the shift register is a single register located in the SCK clock domain, and the transmit (shift-out) portion is the buffer location pointed to by the SPIRDPTR. When a full word / byte is clocked into the SPI shift register, it is copied into the location pointed by SPIWRPTR.

**Figure 32-43.** FIFO Overview




The interrupts and DMA triggers are generated according to FIFO threshold settings in the Control C register (CTRLC.TXTRHOLD, CTRLC.RXTRHOLD).

The Data Register Empty interrupt flag, and the DMA TX trigger respectively, are generated when the available place in the TX FIFO is equal or higher than the threshold value defined by the


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

CTRLC.TXTRHOLD settings. The Transfer complete interrupt is generated when the TX FIFO is empty and the last bit is shifted out.

The Receive Complete interrupt flag, and the DMA RX trigger respectively, are generated when the number of bytes present in the RX FIFO equals or is higher than the threshold value defined by the CTRLC.RXTRHOLD settings. The ERROR interrupt flag is generated when both the RX shifter and the RX FIFO are full.

The FIFO is fully accessible if the SERCOM is halted, by writing the corresponding CPU FIFO pointer in the FIFOPTR register. The RX or TX FIFO can be individually cleared, by setting the respective FIFO Clear bit in the Control B register (CTRLB.FIFOCLR). The FIFO Clear must be written before data transfer begins. Writing the FIFO Clear bits while a frame is in progress will produce unpredictable results.

**Pointer Operation in Host Mode**

In Host mode, the transmit / receive sequence is started by the CPU writing one or more transmit words into the TX FIFO. The CPUWRPTR is incremented by 1 every time the CPU writes a word to the memory array. As soon as the CPUWRPTR becomes not-equal to SPIRDPTR (FIFOSPACE.TXSPACE != 0), the SPI transmits the data pointed to by the SPIRDPTR through HOCI. When a complete word is shifted in, the SPIRDPTR is compared to CPUWRPTR. If they are not equal, SPIRDPTR is incremented, another byte / word is shifted in/out, and so on. When the CPU completes a read from the RX FIFO location (FIFOSPACE.RXSPACE != 0), the CPURDPTR pointer is incremented. When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The module will not respond to SCK transitions while BUFOVF is '1', effectively disabling the module until software reads DATA register.

All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then rolls over to '0'. CPURDPTR will not increment past SPIWRPTR. In other words, if CPURDPTR = SPIWRPTR, and the CPU attempts another read, the pointer will stay at the value of SPIWRPTR.

**Pointer Operation in Client Mode**

In Client mode, the transmit / receive sequence is started by the SPI receiving an SCK clock pulse. As soon as an SCK pulse is received, the SPI transmits the data pointed to by the SPIRDPTR. When a complete data is shifted in, the SPIRDPTR / SPIWRPTR are incremented, another data is shifted in/out, and so on. The newly received data is written to the RX FIFO location pointed to by SPIWRPTR. The CPUWRPTR is incremented by one every time the CPU writes new data to the TX FIFO memory array. If the CPUWRPTR is pointing to location n, and SPIWRPTR is pointing to location n, and a wrap is not detected, the CPUWRPTR will auto-increment to SPIWRPTR, to keep up with SPIWRPTR. This is so that the next data to be transmitted will be placed in the correct position in the storage element. The CPU can read data from RX FIFO as long as FIFOSPACE.RXSPACE != 0. When both RX shifter and RX FIFO are full, the Buffer Overflow status bit is set (STATUS.BUFOVF) and optional ERROR interrupt is generated. The module will not respond to SCK transitions while BUFOVF is '1', effectively disabling the module until software reads RX FIFO. All pointers increment to their maximum value, dictated by CTRLC.DATA32B bit, and then roll over to '0'. CPURDPTR will not increment past SPIWRPTR. In other words, if CPURDPTR = SPIWRPTR, and the CPU attempts another read, the pointer will stay at the value of SPIWRPTR.

#### 32.6.4.3.11. Debug Operation

When the CPU is halted in Debug mode, this peripheral will continue normal operation. If the peripheral is configured to require periodic 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.


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

### 32.6.4.4. DMA Requests and Interrupts

**Table 32-10.** Module Request for SERCOM SPI

| Condition | DMA Request | Interrupt Request |
|---|---|---|
| Standard (DRE): Data Register Empty<br>FIFO (DRE): at least TXTRHOLD locations in TX FIFO are empty | Yes<br>(request cleared when data is written) | Yes |
| Standard (RXC): Receive Complete<br>FIFO (RXC): at least RXTRHOLD data available in RX FIFO, or a last word available and length frame reception completed. | Yes<br>(request cleared when data is read) | Yes |
| Standard (TXC): Transmit Complete<br>FIFO (TXC): Transmit Complete and TX FIFO is empty | N/A | Yes |
| Client Select low (SSL) | N/A | Yes |
| Error (ERROR) | N/A | Yes |

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.

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)
- Client Select Low (SSL)
- Error (ERROR)

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 “Device DMAC – Direct Memory Access Controller” for details.

### 32.6.4.5. Sleep Mode Operation

The behavior in Sleep mode depends on the host/client configuration and the Run In Standby bit in the Control A register (CTRLA.RUNSTDBY):

- Host operation, CTRLA.RUNSTDBY=1: The peripheral clock GCLK_SERCOM_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_SERCOM_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

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.

### 32.6.4.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some SERCOM registers need to be synchronized when written ("Write-Synchronized") or read ("Read-Synchronized").

The following bits are synchronized when written:

- The Software Reset bit in the CTRLA register (CTRLA.SWRST)


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

- The Enable bit in the CTRLA register (CTRLA.ENABLE)
- The Receiver Enable bit in the CTRLB register (CTRLB.RXEN)

**Note:** CTRLB.RXEN is write-synchronized somewhat differently than other registers. See the “CTRLB” register for details.

Required write synchronization is denoted by the "Write-Synchronized" property in the register description. If a write-synchronized register is written while a synchronization is ongoing, an APB (Advanced Peripheral Bus) Error will be generated.

## 32.7. SERCOM I²C

### 32.7.1. Overview

The Inter-Integrated Circuit (I²C) interface is one of the available modes in the Serial Communication Interface module (SERCOM).

The I²C interface uses the SERCOM transmitter and receiver configured as shown in the Block Diagram. 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. The I²C host mode uses the SERCOM baud-rate generator, while the I²C client mode uses the SERCOM address match logic.

In addition, the I²C Host uses the SERCOM baud-rate generator, while the I²C Client uses the SERCOM address match logic .

### 32.7.2. 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


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

**Table 32-11.** SERCOM I²C Features Depending on Package

| Feature | SERCOM I²C Features | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0 | | 1 | | 2 / 3 | | 4 / 5 |
| I²C speed | HS | | FM+ | HS | FM+ | | - |
| IOSET | 1 | 2 | 1 / 2 | 3 | 1 | 2 | - |
| 5V tolerant support | - | - | Yes | - | - | - | - |
| secure capable | - | - | Yes | - | - | - | - |
| available on 48 pins | x | - | x | - | x | - | - |
| available on 64 pins | x | - | x | - | x | - | - |
| available on 100 pins | x | x | x | x | x | x | - |

### 32.7.3. Block Diagram

**Figure 32-44.** I²C Single-Host Single-Client Interconnection

### 32.7.4. Clocks

The SERCOM bus clock (CLK_SERCOM_APB) can be enabled and disabled in the Main Clock Controller. Refer to the "Peripheral Clock Masking" section for details and default status of this clock.

Two generic clocks are used by SERCOM, GCLK_SERCOMx_CORE and CLK_SERCOMx_SLOW. The core clock (GCLK_SERCOMx_CORE) can clock the I²C 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 I²C.

**Note:** The "GCLK_SERCOMn_SLOW" clock is ONLY used for I2C mode exclusively.

These generic clocks are asynchronous to the bus clock (CLK_SERCOM_APB). Due to this asynchronicity, writes to certain registers will require synchronization between the clock domains.

### 32.7.5. Functional Description

#### 32.7.5.1. 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 following figure illustrates the possible transaction formats. These symbols will be used in the following descriptions.


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

**Figure 32-45.** Transaction Diagram Symbols

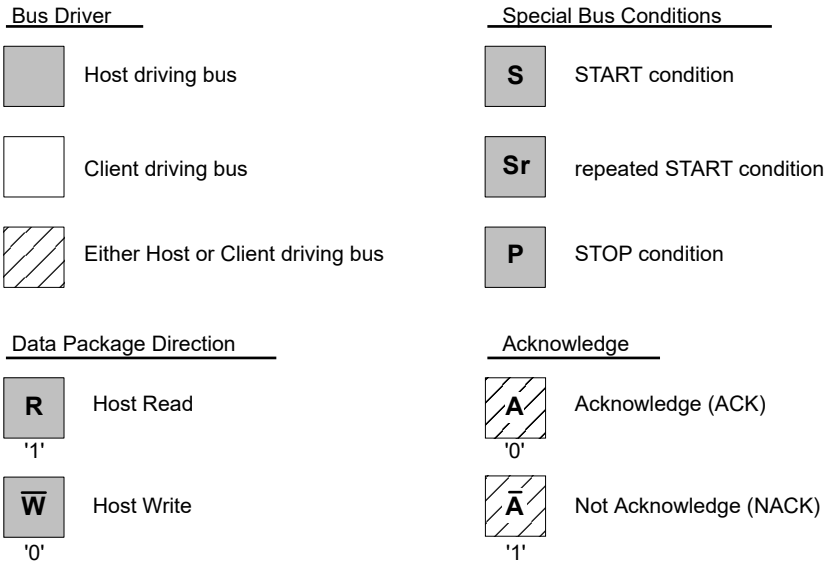
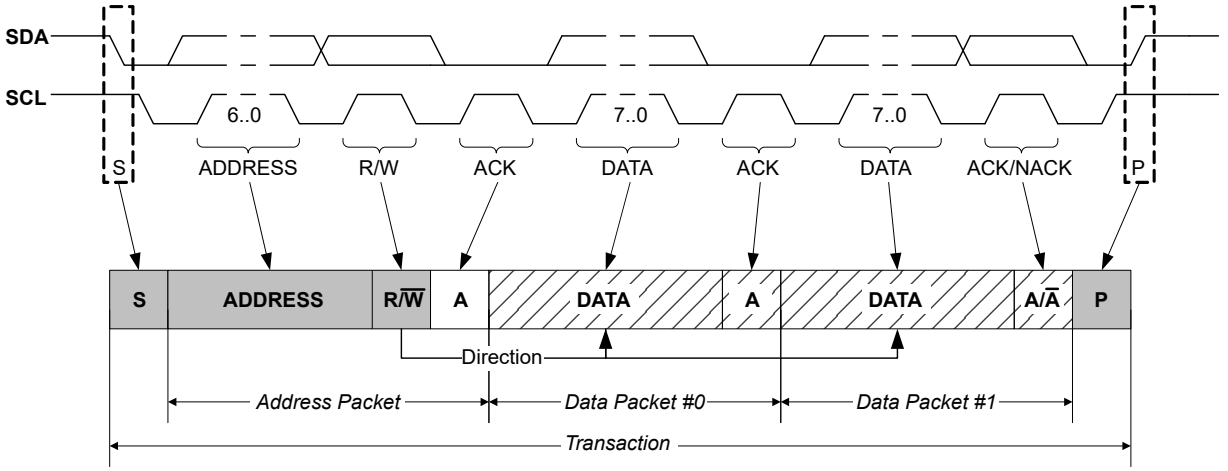



**Figure 32-46.** Basic I²C Transaction Diagram




## 32.7.5.2. Basic Operation

### 32.7.5.2.1. 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 E register (CTRLE), except Acknowledge Action (CTRLE.ACKACT) and Command(CTRLE.CMD) bits
- Control D register (CTRLD) Baud register (BAUD) Address register (ADDR) in Client operation.
- 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


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

the disabling. Enable-protection is denoted by the "Enable-Protection" property in the register description

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.LOWTOUT).
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).
   b. Set the Address and Address Mask value in the Address register (ADDR.ADDR and ADDR.ADDRMASK) according to the address configuration.

#### 32.7.5.2.2. 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.

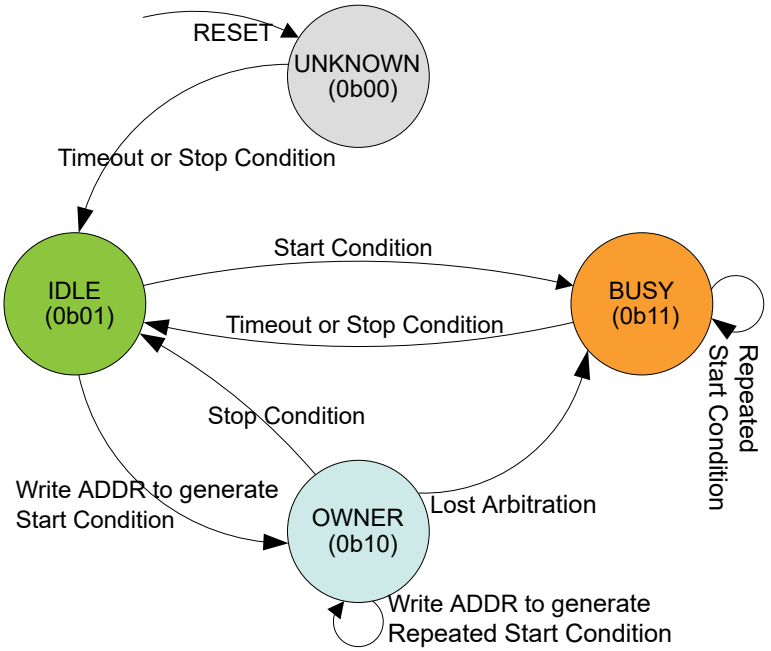
Writing ‘1’ to the Software Reset bit in the Control A register (CTRLA.SWRST) will reset all registers of this peripheral to their initial states, except the DBGCTRL register, and the peripheral is disabled.

#### 32.7.5.2.3. 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 the following figure. 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.


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

**Figure 32-47.** 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.
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').

### 32.7.5.2.4. 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 (CTRLB.SMEN).

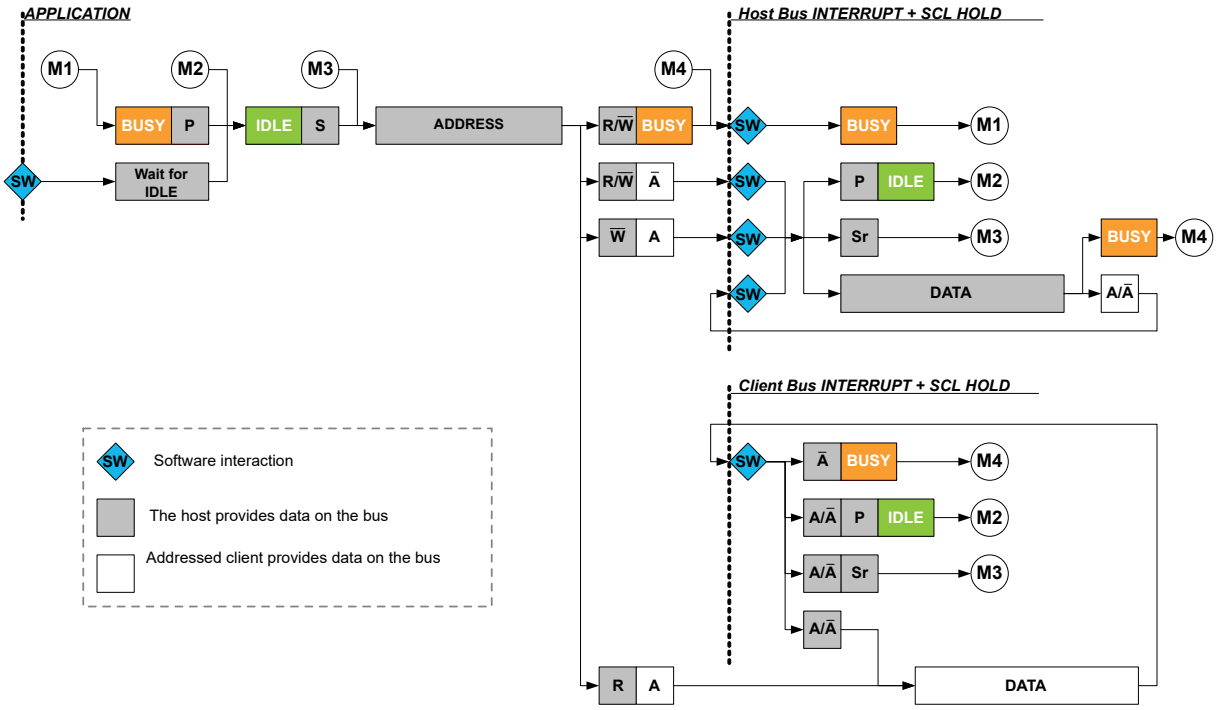

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The I²C host has two interrupt strategies.

When SCL Stretch Mode (CTRLD.SCLSM) is '0', SCL is stretched before or after the Acknowledge bit . In this mode the I²C host operates according to the following diagram. 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.

**Figure 32-48.** I²C Host Behavioral Diagram (SCLSM=0)

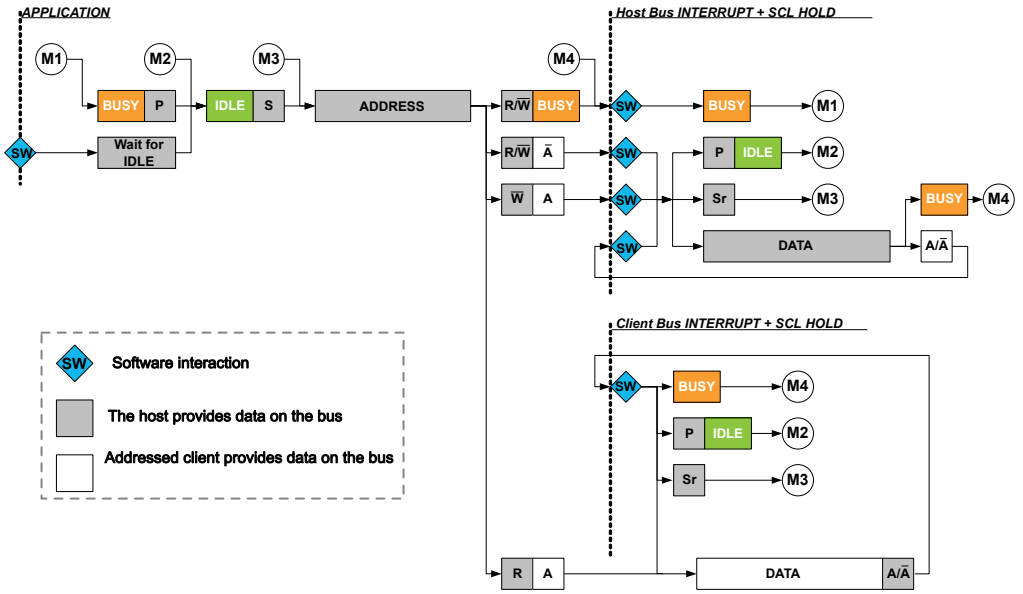



In the second strategy (CTRLD.SCLSM=1), interrupts only occur after the ACK bit, as in the following diagram. This strategy can be used when it is not necessary to check DATA before acknowledging.

**Note:** I²C High-speed (*Hs*) mode requires CTRLD.SCLSM=1.


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

**Figure 32-49.** I²C Host Behavioral Diagram (SCLSM=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

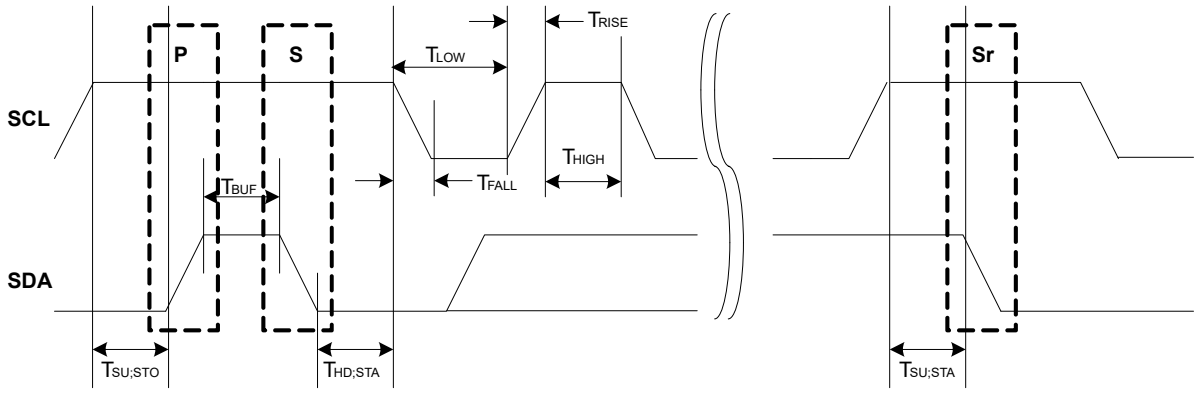
The Host clock configuration for *Sm, Fm,* and *Fm+* are described in *Clock Generation (Standard-Mode, Fast-Mode, and Fast-Mode Plus)*. For *Hs*, refer to *Host Clock Generation (High-Speed Mode)*.

### 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 (TLOW) and high (THIGH) times are determined by the Baud Rate register (BAUD), while the rise (TRISE) and fall (T FALL) times are determined by the bus topology. Because of the wired-AND logic of the bus, TFALL will be considered as part of TLOW. Likewise, TRISE will be in a state between TLOW and THIGH until a high state has been detected.

**Figure 32-50.** SCL Timing





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

The following parameters are timed using the SCL low time period TLOW. This comes from the Host Baud Rate Low bit group in the Baud Rate register (BAUD.BAUDLOW). When BAUD.BAUDLOW=0, the Host Baud Rate bit group in the Baud Rate register (BAUD.BAUD) determines it.

- **TLOW:** Low period of SCL clock
- **TSU;STO:** Set-up time for stop condition
- **TBUF:** Bus free time between stop and start conditions
- **THD;STA:** Hold time (repeated) start condition
- **TSU;STA:** Set-up time for repeated start condition
- **THIGH:** Timed using the SCL high time count from BAUD.BAUD
- **TRISE:** Determined by the bus impedance; for internal pull-ups
- **TFALL:** 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 I²C 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.

**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:


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

$$f_{SCL} = \frac{f_{GCLK}}{2 + HS\,BAUD + HSBAUDLOW}$$

**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.

**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 5.6.1 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**

If the I²C host receives an ACK 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


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

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.

### Transmitting Data Packets

When an address packet with direction Host Write is 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.

### 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.

### 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 CTRLE.ACKACT. At this point, CTRLE.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.

### High-Speed Mode

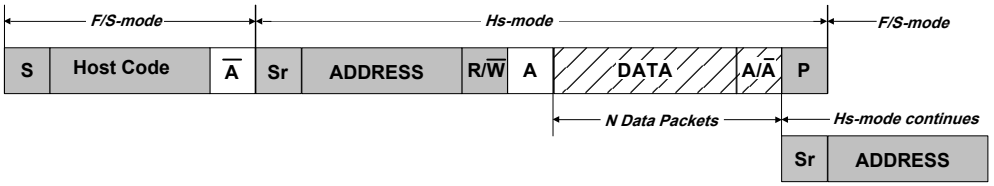
High-speed transfers are a multi-step process, for further information, refer to the following figure.

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 meantime, 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.


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

**Figure 32-51.** High Speed Transfer



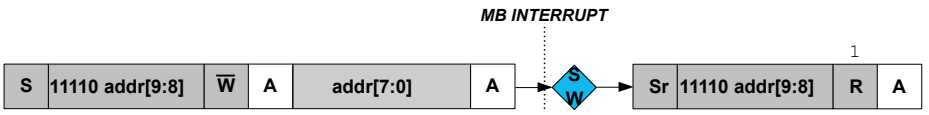

Transmitting in High-speed mode requires the I²C host to be configured in High-speed mode (CTRLD.SPEED=0x2) and the SCL Clock Stretch mode (CTRLD.SCLSM) bit set to '1'.

### 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 32-52.** 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.

## 32.7.5.2.5. 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 D register (CTRLD.SMEN).

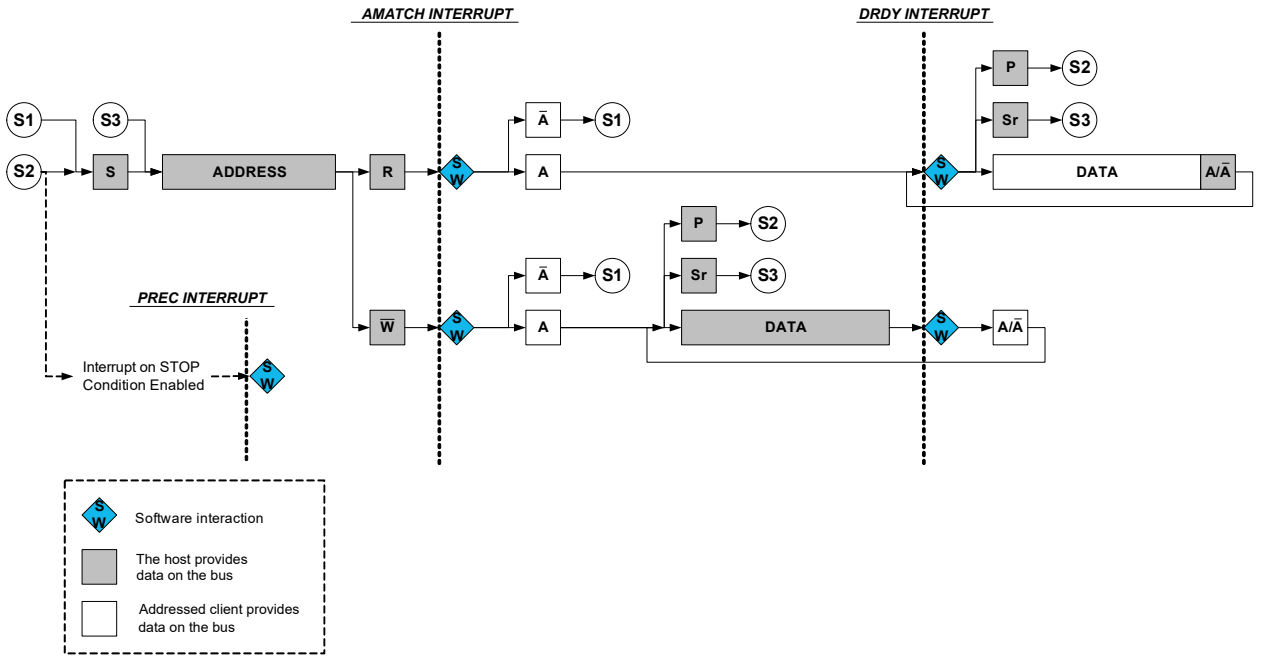
The I²C client has two interrupt strategies.

When SCL Stretch Mode bit (CTRLD.SCLSM) is '0', SCL is stretched before or after the acknowledge bit. In this mode, the I²C client operates according to the following figure. The circles labeled "S*n*" (S1, S2..) 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 client operation throughout the document.


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

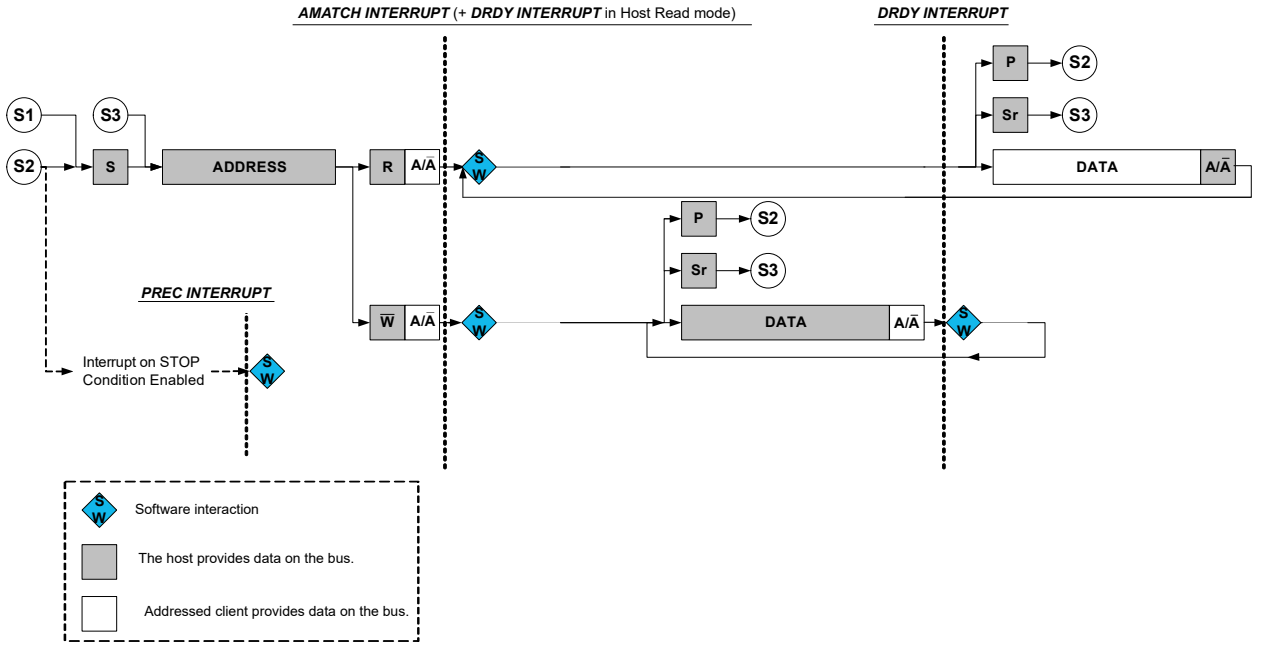
**Figure 32-53.** I²C Behavioral Diagram (SCLSM=0)




In the second strategy (CTRLD.SCLSM=1), interrupts only occur after the ACK bit is sent as shown in Client Behavioral Diagram (SCLSM=1). This strategy can be used when it is not necessary to check DATA before acknowledging. For host reads, an address and data interrupt will be issued simultaneously after the address acknowledge. However, for host writes, the first data interrupt will be seen after the first data byte has been received by the client and the acknowledge bit has been sent to the host.

**Note:** For I²C High-speed mode (*Hs*), SCLSM=1 is required.

**Figure 32-54.** I²C Client Behavioral Diagram (SCLSM=1)





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

### Receiving Address Packets (SCLSM = 0)

When CTRLD.SCLSM=0, the I²C client stretches the SCL line according to the I²C Behavioral Diagram (SCLSM0). 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 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 E register (CTRLE.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 CTRLE.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 CTRLE.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 CTRLE.ACKACT bit.

### Receiving Address Packets (SCLSM=1)

When SCLSM=1, the I²C client will stretch the SCL line only after an ACK, see 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 E register (CTRLE.ACKACT) will be sent and the Address Match bit in the Interrupt Flag register


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

(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.

### 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 CTRLE.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.

### High-Speed Mode

When the I²C client is configured in High-speed mode (*Hs*, CTRLD.SPEED=0x2) and CTRLD.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.

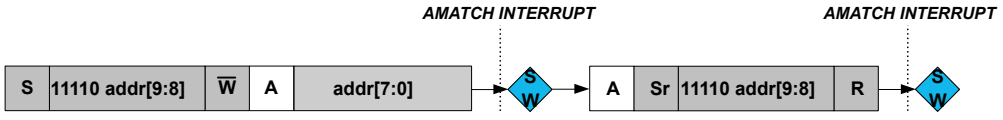
### 10-bit Addressing (Client)

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 32-55.** 10-bit Addressing





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

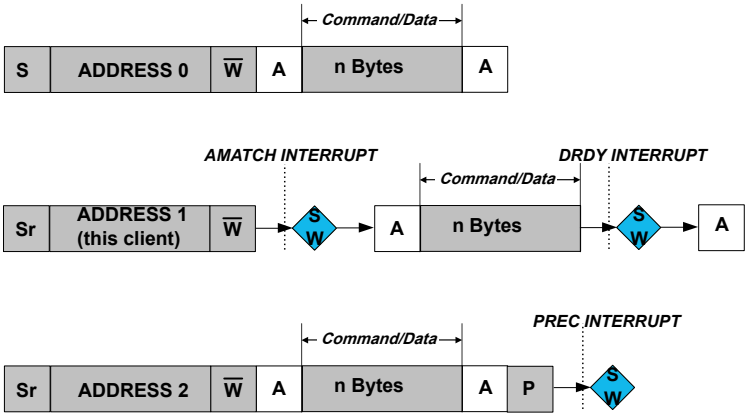
### PMBus Group Command

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.

**Figure 32-56.** PMBus Group Command Example




## 32.7.5.3. Additional Features

### 32.7.5.3.1. SMBus

The $I^2C$ 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 KHz oscillator. The $I^2C$ 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 CTRLD.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 CTRLD.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 CTRLD.MEXTTOEN.

The SMBus-compatible logic levels are enabled by writing the SMBus Input Buffer Enable bit in Control A register (CTRLD.SMBUFEN).

### 32.7.5.3.2. Smart Mode

The $I^2C$ interface has a Smart mode that simplifies application code and minimizes the user interaction needed to adhere to the $I^2C$ protocol. The Smart mode accomplishes this by

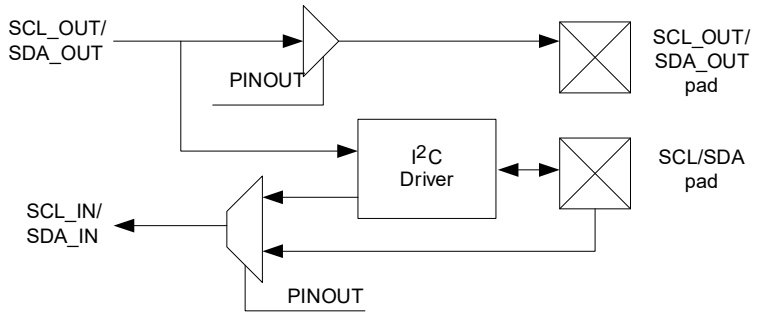

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

automatically issuing an ACK or NACK (based on the content of CTRLE.ACKACT) as soon as DATA.DATA is read.

#### 32.7.5.3.3. 4-Wire Mode

Writing a '1' to the Pin Usage bit in the Control D register (CTRLD.PINOUT) will enable 4-Wire mode operation. In this mode, the internal $I^2C$ tri-state drivers are bypassed, and an external $I^2C$ compliant tri-state driver is needed when connecting to an $I^2C$ bus.

**Figure 32-57.** $I^2C$ Pad Interface




#### 32.7.5.3.4. 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 CTRLE.CMD or ADDR.ADDR.

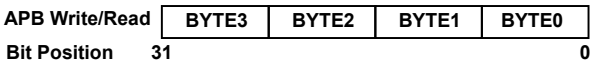
#### 32.7.5.3.5. 32-bit Extension

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.

The following figure 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.

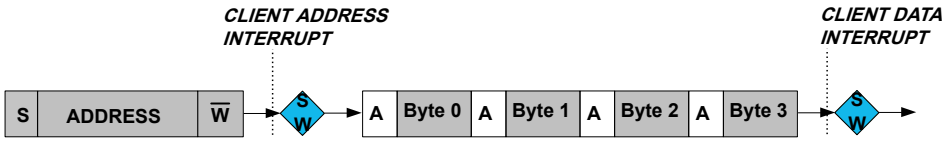
**Figure 32-58.** 32-bit Extension Byte Ordering




#### 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 32-59.** 32-bit Extension Client Operation





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

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 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 (CTRLD.SCLSM=1) and the transaction is a host write, the selected Acknowledge Action (CTRLE.ACKACT) will only be used to ACK/NACK each 4th byte. All other bytes are ACKed. This allows the user to write CTRLE.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 (CTRLD.SCLSM=1) and the transaction is a host read, the selected Acknowledge Action (CTRLE.ACKACT) will only be used to ACK/NACK each 4th Byte. All other bytes are ACKed. This allows the user to set CTRLE.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.

### 32.7.5.4. DMA and Interrupts

This chapter provides DMA and interrupt conditions when the FIFO is disabled. For details when the FIFO is enabled, refer to FIFO Support.

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 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.


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

**Table 32-12.** Module Request for SERCOM I²C Client

| Condition | Request DMA | Interrupt | Event |
|---|---|---|---|
| Data needed for transmit (TX)<br>(Client Transmit mode) | Yes<br>(request cleared when data is written) | | N/A |
| Data received (RX)<br>(Client Receive mode) | Yes<br>(request cleared when data is read) | | |
| Data Ready (DRDY) | | Yes | |
| Address Match (AMATCH) | | Yes | |
| Stop received (PREC) | | Yes | |
| TX FIFO Empty (TXFE) | | Yes | |
| RX FIFO Full (RXFF) | | Yes | |
| Error (ERROR) | | Yes | |

**Table 32-13.** Module Request for SERCOM I²C Host

| Condition | Request DMA | Interrupt | Event |
|---|---|---|---|
| Data needed for transmit (TX)<br>(Host Transmit mode) | Yes<br>(request cleared when data is written) | | N/A |
| Data needed for transmit (RX)<br>(Host Transmit mode) | Yes<br>(request cleared when data is read) | | |
| Host on Bus (MB) | | Yes | |
| Stop received (SB) | | Yes | |
| TX FIFO Empty (TXFE) | | Yes | |
| RX FIFO Full (RXFF) | | Yes | |
| Error (ERROR) | | Yes | |

#### 32.7.5.4.1. DMA Operation

Smart mode must be enabled for DMA operation in the Control B register by writing CTRLB.SMEN=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): If the FIFO is disabled, the request is set when host write data is received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached (CTRLC.RXTRHOLD). The request is cleared when DATA is read.
- Read data needed for transmit (TX): If the FIFO is disabled, the request is set when data is needed for a host read operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

**Host DMA**

When using the I²C host with DMA, the ADDR register must be written with the desired address (ADDR.ADDR) and transaction length (ADDR.LEN), with 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:


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

- Read data received (RX): If the FIFO is disabled, the request is set when host read data is received. If the FIFO is enabled, the request is set when the RX FIFO threshold is reached. The request is cleared when DATA is read.
- Write data needed for transmit (TX): If the FIFO is disabled, the request is set when data is needed for a host write operation. If the FIFO is enabled, the request is set when the TX FIFO threshold is reached (CTRLC.TXTRHOLD). The request is cleared when DATA is written.

#### 32.7.5.4.2. Interrupts

The $I^2C$ client has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- RX FIFO Full (RXFF)
- TX FIFO Empty (TXFE)
- Data Ready (DRDY)
- Address Match (AMATCH)
- Stop Received (PREC)

The $I^2C$ host has the following interrupt sources. These are asynchronous interrupts. They can wake-up the device from any Sleep mode:

- Error (ERROR)
- RX FIFO Full (RXFF)
- TX FIFO Empty (TXFE)
- 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.

### 32.7.5.5. Sleep Mode Operation

#### 32.7.5.5.1. $I^2C$ 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.

#### 32.7.5.5.2. $I^2C$ 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.

### 32.7.5.6. Synchronization

Due to asynchronicity between the main clock domain and the peripheral clock domains, some SERCOM registers need to be synchronized when written ("Write-Synchronized") or read ("Read-Synchronized").


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

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 CTRLE register (CTRLE.CMD)
- FIFO Clear bits in CTRLE register (CTRLE.FIFOCLR)
- 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.

If a write-synchronized register is written while a synchronization is ongoing, a Bus Error exception will be generated.

## 32.8. Peripheral Dependencies

**Table 32-14.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| SERCOM0 | 0x44808000 | 60 : ERROR<br>61 : RXBRK<br>62 : DRE/ PREC/MB<br>63 : TXC/ AMATCH/SB<br>64 : RXC/DRDY<br>65 : RXS/SSL/ TXFE<br>66 : CTSIC/RXFF | **APB:**<br>CLKMSK[1]<br>MASK4 | **SLOW:**<br>PCHCTRL[20]<br>**CORE:**<br>PCHCTRL[21] | 26<br>STATUS0<br>PERID26 | 4 : RX<br>5 : TX | | PD_CORE_SW |
| SERCOM1 | 0x4480A000 | 67 : ERROR<br>68 : RXBRK<br>69 : DRE/ PREC/MB<br>70 : TXC/ AMATCH/SB<br>71 : RXC/DRDY<br>72 : RXS/SSL/ TXFE<br>73 : CTSIC/RXFF | **APB:**<br>CLKMSK[1]<br>MASK5 | **SLOW:**<br>PCHCTRL[20]<br>**CORE:**<br>PCHCTRL[22] | 27<br>STATUS0<br>PERID27 | 6 : RX<br>7 : TX | | PD_CORE_SW |
| SERCOM2 | 0x4480C000 | 74 : ERROR<br>75 : RXBRK<br>76 : DRE/ PREC/MB<br>77 : TXC/ AMATCH/SB<br>78 : RXC/DRDY<br>79 : RXS/SSL/ TXFE<br>80 : CTSIC/RXFF | **APB:**<br>CLKMSK[1]<br>MASK6 | **SLOW:**<br>PCHCTRL[20]<br>**CORE:**<br>PCHCTRL[23] | 28<br>STATUS0<br>PERID28 | 8 : RX<br>9 : TX | | PD_CORE_SW |


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

**Table 32-14.** Peripheral Dependencies (continued)

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| SERCOM3 | 0x4480E000 | 81 : ERROR<br>82 : RXBRK<br>83 : DRE/ PREC/MB<br>84 : TXC/ AMATCH/SB<br>85 : RXC/DRDY<br>86 : RXS/SSL/ TXFE<br>87 : CTSIC/RXFF | **APB:**<br>CLKMSK[1]<br>MASK7 | **SLOW:**<br>PCHCTRL[20]<br>**CORE:**<br>PCHCTRL[24] | 29<br>STATUS0<br>PERID29 | 10 : RX<br>11 : TX | | PD_CORE_SW |
| SERCOM4 | 0x44820000 | 104 : ERROR<br>105 : RXBRK<br>106 : DRE/ PREC/MB<br>107 : TXC/ AMATCH/SB<br>108 : RXC/ DRDY<br>109 : RXS/SSL/ TXFE<br>110 : CTSIC/ RXFF | **APB:**<br>CLKMSK[1]<br>MASK12 | PCHCTRL[28] | 34<br>STATUS1<br>PERID2 | 24 : RX<br>25 : TX | | PD_CORE_SW |
| SERCOM5 | 0x44822000 | 111 : ERROR<br>112 : RXBRK<br>113 : DRE/ PREC/MB<br>114 : TXC/ AMATCH/SB<br>115 : RXC/ DRDY<br>116 : RXS/SSL/ TXFE<br>117 : CTSIC/ RXFF | **APB:**<br>CLKMSK[1]<br>MASK13 | PCHCTRL[29] | 35<br>STATUS1<br>PERID3 | 26 : RX<br>27 : TX | | PD_CORE_SW |


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

## 32.9. Register Summary - USART

| 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 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | | | | | | | GTIME[2:0] | |
| | | 15:8 | | | | | | HDRDLY[1:0] | | BRKLEN[1:0] |
| | | 23:16 | | | | MAXITER[2:0] | | | DSNACK | INACK |
| | | 31:24 | | TXTRHOLD[1:0] | | | RXTRHOLD[1:0] | FIFOEN | | 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 | | | | | | | | |
| 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 |
| 0x31 ... 0x33 | Reserved | | | | | | | | | |
| 0x34 | FIFOSPACE | 7:0 | | | | | | TXSPACE[4:0] | | |
| | | 15:8 | | | | | | RXSPACE[4:0] | | |
| 0x36 | FIFOPTR | 7:0 | | | | | | CPUWRPTR[3:0] | | |
| | | 15:8 | | | | | | CPURDPTR[3:0] | | |


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

### 32.9.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 32-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | Name | Description |
|---|---|---|
| 0x0 | MSB | MSB is transmitted first. |
| 0x1 | LSB | 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.


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

| Value | Name | Description |
|---|---|---|
| 0x0 | ASYNC | Asynchronous communication. |
| 0x1 | SYNC | Synchronous communication. |

**Bits 27:24 – FORM[3:0]** Frame Format

These bits define the frame format.
These bits are not synchronized.

| FORM[3:0] | Description |
|---|---|
| 0x0 | USART frame |
| 0x1 | USART frame with parity |
| 0x2 | LIN Hosst - 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 - 0xF | Reserved |

| Value | Name | Description |
|---|---|---|
| 0x0 | USART | USART frame |
| 0x1 | USARTP | USART frame with parity |
| 0x2 | LINBRKGEN | LIN Master Break and Synck generation |
| 0x4 | LINBRKDET | LIN Slave break detection and auto-baud |
| 0x5 | BRKDET | Break detection and auto-baud with parity |

**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 (Recommended) | 3-4-5 (Recommended) |
| 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 |

| Value | Name | Description |
|---|---|---|
| 0x0 | PAD0 | SERCOM PAD0 is used for data reception |
| 0x1 | PAD1 | SERCOM PAD1 is used for data reception |
| 0x2 | PAD2 | SERCOM PAD2 is used for data reception |
| 0x3 | PAD3 | SERCOM PAD3 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 | | | |


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

| TXPO (continued) | | | | |
|---|---|---|---|---|
| TXPO | TxD Pin Location | XCK Pin Location (When Applicable) | RTS/TE | CTS |
| 0x2 | SERCOM PAD[0] | N/A | SERCOM PAD[2] | SERCOM PAD[3] |
| 0x3 | SERCOM PAD[0] | SERCOM PAD[1] | SERCOM PAD[2] | N/A |

| Value | Name | Description |
|---|---|---|
| 0x0 | PAD0 | PAD[0] = TxD; PAD[1] = XCK |
| 0x2 | PAD2 | PAD[0] = TxD; PAD[2] = RTS; PAD[3] = CTS |
| 0x3 | PAD3 | PAD[0] = TxD; PAD[1] = XCK; PAD[2] = TE |

**Bits 15:13 – SAMPR[2:0]** Sample

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 |

| Value | Name | Description |
|---|---|---|
| 0x0 | ARITHM16X | 16x oversampling using arithmetic baud rate generation |
| 0x1 | FRAC16X | 16x oversampling using fractional baud rate generation |
| 0x2 | ARITHM8X | 8x oversampling using arithmetic baud rate generation |
| 0x3 | FRAC8X | 8x oversampling using fractional baud rate generation |
| 0x4 | ARITHM3X | 3x oversampling using arithmetic baud rate generation |

**Bit 10 – RXINV** Receive Data Invert

This bit controls whether the receive data (RxD) is inverted or not.
**Notes:**

1. Start, parity and stop bit(s) 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).

This bit is not synchronized.

| 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 bit(s) 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).

This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | TxD is not inverted. |
| 1 | TxD is inverted. |


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

**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.
This bit is not synchronized.

| 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 during 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 either Receive Start or Transfer Complete interrupt unless the appropriate ONDEMAND bits are set in the clocking chain. |
| 0x1 | Wake on Receive Start or 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 | Name | Description |
|---|---|---|
| 0x0 | EXTCLK | USART with external clock |
| 0x1 | INTCLK | 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. |

**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 a bus 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.

**Notes:**

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by the hardware.


© 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.

### 32.9.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

Write to this register only when SYNCBUSY.CTRLB = 0, otherwise a bus error will result.

**Table 32-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| 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 |

**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. |

**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.


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

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 '1'). 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 | Name | Description |
|---|---|---|
| 0x0 | EVEN | Even parity |
| 0x1 | ODD | Odd parity |

**Bit 10 – ENC** Encoding Format
This bit selects the data encoding format.
This bit is not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Data is not encoded |
| 0x1 | IRDA | 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.

| Value | Name | Description |
|---|---|---|
| 0x0 | ONE | One stop bit |
| 0x1 | TWO | Two stop bits |


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

**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 |

| Value | Name | Description |
|---|---|---|
| 0x0 | 8BITS | 8-bits character |
| 0x1 | 9BITS | 9-bits character |
| 0x5 | 5BITS | 5-bits character |
| 0x6 | 6BITS | 6-bits character |
| 0x7 | 7BITS | 7-bits character |


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

### 32.9.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 32-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TXTRHOLD[1:0] | | RXTRHOLD[1:0] | | FIFOEN | | DATA32B[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 | 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 31:30 – TXTRHOLD[1:0]** Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

| TXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated as long as the FIFO is not full. |
| 1 | HALF | Interrupt and DMA triggers are generated when half FIFO space is free. |
| 2 | EMPTY | Interrupt and DMA triggers are generated when the FIFO is empty. |
| 3 | - | Reserved |

**Bits 29:28 – RXTRHOLD[1:0]** Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

| RXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated when a DATA is present in the FIFO. |
| 1 | HALF | Interrupt and DMA triggers can be generated only when the FIFO is half-full. |
| 2 | FULL | Interrupt and DMA triggers can be generated only when the FIFO is full. |
| 3 | - | Reserved |

**Bit 27 – FIFOEN** FIFO Enable
This bit enables the FIFO operation.


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

| Value | Description |
|---|---|
| 0 | FIFO operation is disabled |
| 1 | FIFO operation is enabled |

**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. |

**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. |


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

**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 in order 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.

### 32.9.4. Baud Rate Register

**Name:** BAUD
**Offset:** 0x0C
**Reset:** 0x0000
**Property:** Enable-Protected, PAC Write-Protection

**Table 32-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Rate 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 or =3`) 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 "Clock Generation – Baud-Rate Generator" section.
- **Bits 12:0 - BAUD[12:0]: Baud Value**
  These bits control the clock generation, as described in the Clock Generation – Baud-Rate Generator section.


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

### 32.9.5. Receive Pulse Length

**Name:** RXPL
**Offset:** 0x0E
**Reset:** 0x00
**Property:** Enable-Protected, PAC Write-Protection

**Table 32-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 $SE_{per}$:
$PULSE \geq (\mathrm{RXPL} + 1) \cdot SE_{per}$


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

### 32.9.6. 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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

**Table 32-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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** Combined Error Interrupt Disable
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.

**Bit 5 – RXBRK** Break Received Interrupt Disable
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.

**Bit 4 – CTSIC** Clear To Send Input Change Interrupt Disable
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.

**Bit 3 – RXS** Receive Start Interrupt Disable
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.

**Bit 2 – RXC** Receive Complete Interrupt Disable
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.

**Bit 1 – TXC** Transmit Complete Interrupt Disable
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.

**Bit 0 – DRE** Data Register Empty Interrupt Disable
Writing '0' to this bit has no effect.


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

Writing '1' to this bit will clear the Data Register Empty Interrupt Enable bit, which disables the Data Register Empty interrupt.


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

### 32.9.7. 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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

**Table 32-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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** Combined 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** Break Received 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. |


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

| Value | Description |
|---|---|
| 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. |
| 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.


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

### 32.9.8. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** -

**Note:** Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 32-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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** Combined Error Interrupt
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** Break Received Interrupt
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 Interrupt
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 Interrupt
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 Interrupt
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.


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

**Bit 1 – TXC** Transmit Complete Interrupt
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 Interrupt
This flag is cleared by writing new data to DATA.
This flag is set when DATA is empty and ready to be written.
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.

### 32.9.9. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** -

**Table 32-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

| Value | Description |
|---|---|
| 0 | No collisions detected |
| 1 | Collision detected |

**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 = 0x4 or 0x5) 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.


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

| Value | Description |
|---|---|
| 0 | No inconsistent sync fields |
| 1 | Inconsistent sync field received |

**Bit 3 – CTS** Clear To Send

This bit indicates the current level of the CTS pin when flow control is enabled (CTRLA.TXPO).

| Value | Description |
|---|---|
| 0 | CTS low |
| 1 | CTS high |

**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.
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.

| Value | Description |
|---|---|
| 0 | No buffer overflow |
| 1 | Buffer Overflow |

**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.

| Value | Description |
|---|---|
| 0 | No Frame Errors |
| 1 | Frame Error received |

**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.

## 32.9.10. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** -

**Table 32-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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, a bus 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, a bus error will be generated.


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

| 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. 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.

### 32.9.11. Receive Error Count Register

**Name:** RXERRCNT
**Offset:** 0x20
**Reset:** 0x00
**Property:** Read-Synchronized

**Table 32-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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


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

### 32.9.12. Length

**Name:** LENGTH
**Offset:** 0x22
**Reset:** 0x0000
**Property:** Write-Synchronized

Write to this register only when SYNCBUSY.LENGTH = 0, otherwise a bus error will result.

**Table 32-26.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Length counter is disabled |
| 0x1 | TX | Length counter is enabled for transmit |
| 0x2 | RX | Length counter is enabled for receive |

**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.

### 32.9.13. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** -

**Table 32-27.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Value

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.

### 32.9.14. Debug Control

**Name:** DBGCTRL
**Offset:** 0x30
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 32-28.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 0 – DBGSTOP** Debug 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.

### 32.9.15. FIFO Space

**Name:** FIFOSPACE
**Offset:** 0x34
**Reset:** 0x0000
**Property:** -

**Table 32-29.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | RXSPACE[4:0] | | | | |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

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

**Bits 12:8 – RXSPACE[4:0]** Rx FIFO Filled Space
These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

**Bits 4:0 – TXSPACE[4:0]** Tx FIFO Empty Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).


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

### 32.9.16. FIFO CPU Pointers

**Name:** FIFOPTR
**Offset:** 0x36
**Reset:** 0x0000
**Property:** -

This register provides a copy of internal CPU TX and RX FIFO pointers.

**Table 32-30.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPURDPTR[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPUWRPTR[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 11:8 – CPURDPTR[3:0]** CPU FIFO Read Pointer
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

**Bits 3:0 – CPUWRPTR[3:0]** CPU FIFO Write Pointer
These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.


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

## 32.10. Register Summary - SPI

| 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 | FIFOCLR[1:0] | | | | | | RXEN | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | | | ICSPACE[5:0] | | | | | |
| | | 15:8 | | | | | IGNTUR | FSPOL | FSLEN | FSES |
| | | 23:16 | | | | | | | FMODE | FRMEN |
| | | 31:24 | TXTRHOLD[1:0] | | RXTRHOLD[1:0] | | FIFOEN | | | 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 | | | | | TUR | 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] | | | | | | | |
| 0x2A ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | DBGCTRL | 7:0 | | | | | | | | DBGSTOP |
| 0x31 ... 0x33 | Reserved | | | | | | | | | |
| 0x34 | FIFOSPACE | 7:0 | | | | TXSPACE[4:0] | | | | |
| | | 15:8 | | | | RXSPACE[4:0] | | | | |
| 0x36 | FIFOPTR | 7:0 | | | | | CPUWRPTR[3:0] | | | |
| | | 15:8 | | | | | CPURDPTR[3:0] | | | |


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

### 32.10.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

**Table 32-31.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |


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

| CPHA (continued) | | | | |
|---|---|---|---|---|
| Mode | CPOL | CPHA | Leading Edge | Trailing Edge |
| 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. |
| 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 HICO.
In Client operation, DI is HOCI.
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 HOCI.
In Client operation, DO is HICO.
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.


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

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. |

**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.

### 32.10.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected, Write-Synchronized

Write to this register only when SYNCBUSY.CTRLB = 0, otherwise a bus error will result.

**Table 32-32.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FIFOCLR[1:0] | | | | | | RXEN | |
| Access | R/W | R/W | | | | | R/W | |
| Reset | 0 | 0 | | | | | 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 |

**Bits 23:22 – FIFOCLR[1:0]** FIFO Clear
When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.CTRLB = 0.
These bits are not enable-protected.

| FIFOCLR[1:0] | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | TXFIFO | Clear TX FIFO |
| 0x2 | RXFIFO | Clear RX FIFO |
| 0x3 | BOTH | Clear both TX/RX FIFO |

**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'.


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

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.

| 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 |

**Bit 13 – MSSEN** Host SPI Select Enable

This bit enables hardware SPI Select ($\overline{SS}$) control.

| 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.

| 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.

**Bits 2:0 – CHSIZE[2:0]** Character Size

| 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.

### 32.10.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 32-33.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TXTRHOLD[1:0] | | RXTRHOLD[1:0] | | FIFOEN | | | DATA32B |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | FMODE | FRMEN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | IGNTUR | FSPOL | FSLEN | FSES |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| 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 |

**Bits 31:30 – TXTRHOLD[1:0]** Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

| TXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated as long as the FIFO is not full. |
| 1 | HALF | Interrupt and DMA triggers are generated when half FIFO space is free. |
| 2 | EMPTY | Interrupt and DMA triggers are generated when the FIFO is empty. |
| 3 | - | Reserved |

**Bits 29:28 – RXTRHOLD[1:0]** Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

| RXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated when a DATA is present in the FIFO. |
| 1 | HALF | Interrupt and DMA triggers can be generated only when the FIFO is half-full. |
| 2 | FULL | Interrupt and DMA triggers can be generated only when the FIFO is full. |
| 3 | - | Reserved |

**Bit 27 – FIFOEN** FIFO Enable
This bit enables the FIFO operation.


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

| Value | Description |
|---|---|
| 0 | FIFO operation is disabled |
| 1 | FIFO operation is enabled |

**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 |

**Bit 17 – FMODE** Frame Mode

This bit defines the Frame Mode.

| FMODE | Name | Description |
|---|---|---|
| 0 | Host | Frame Host |
| 1 | Client | Frame Client |

**Bit 16 – FRMEN** Frame Mode Enable

This bit enables the SPI Frame Mode operation.

| FRMEN | Name | Description |
|---|---|---|
| 0 | DISABLE | Frame Mode Disabled |
| 1 | ENABLE | Frame Mode Enabled |

**Bit 11 – IGNTUR** Ignore Transmit Underrun

This bit controls when the underrun conditions in framed mode must be ignored.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | When a new FSYNC is detected, zero-bytes frames will be sent while underrun condition detection is not cleared. |
| 1 | When a new FSYNC is detected, DATA will be transmitted. |

**Bit 10 – FSPOL** Frame Synch Polarity

This bit defines the valid Frame Synch polarity.

| FSPOL | Name | Description |
|---|---|---|
| 0 | HIGH | VCC-level valid polarity |
| 1 | LOW | Ground-level valid polarity |

**Bit 9 – FSLEN** Frame Synch Length

This bit defines the Frame Synch duration.

| FSLEN | Name | Description |
|---|---|---|
| 0 | STROBE | One SCK pulse |
| 1 | LEVEL | One frame duration valid level |

**Bit 8 – FSES** Frame Sync Edge Select

This bit controls when the frame sync pulse edge must be generated.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Frame synchronization pulse (idle-to-active edge) precedes the first bit clock. |
| 1 | Frame synchronization pulse (idle-to-active edge) coincides with the first bit clock. |


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

**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.

### 32.10.4. Baud Rate Register

**Name:** BAUD
**Offset:** 0x0C
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

**Table 32-34.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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*.


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

### 32.10.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00
**Property:** PAC Write-Protection

This register allows the user to disable an interrupt without read-modify-write operation. Changes in this register will also be reflected in the Interrupt Enable Set register (INTENSET).

**Table 32-35.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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. |


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

**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.

### 32.10.6. 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).

**Table 32-36.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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. |


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

**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.

### 32.10.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** -

**Table 32-37.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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. Errors that will set this flag have corresponding Status flags in the STATUS register. The TUR error,, 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.


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

Writing '1' to this bit has no effect.


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

### 32.10.8. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** -

**Table 32-38.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | LENERR | | | |
| Access | | | | | R/W | | | |
| Reset | | | | | 0 | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | TUR | BUFOVF | | |
| Access | | | | | R/W | R/W | | |
| Reset | | | | | 0 | 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 3 – TUR** Frame Transmit Underrun
This bit is cleared by writing '1' to the bit or by disabling the receiver. This bit is set when an underflow condition is detected in frame mode. Writing '0' to this bit has no effect. Writing '1' to this bit sends a request to clear it. It will be actually cleared after the DATA buffer is flushed.

| Value | Description |
|---|---|
| 0 | No Underrun has occurred. |
| 1 | A Underrun 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.

### 32.10.9. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** -

**Table 32-39.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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. |


© 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. 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. |

**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.

### 32.10.10. Length

**Name:** LENGTH
**Offset:** 0x22
**Reset:** 0x0000
**Property:** PAC Write-Protection, Write-Synchronized

**Table 32-40.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 32.10.11. Address

**Name:** ADDR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 32-41.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Value
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.

## 32.10.12. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x0000
**Property:** –

**Table 32-42.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Value

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.

### 32.10.13. Debug Control

**Name:** DBGCTRL
**Offset:** 0x30
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 32-43.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 0 – DBGSTOP** Debug 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.

### 32.10.14. FIFO Space

**Name:** FIFOSPACE
**Offset:** 0x34
**Reset:** 0x0000
**Property:** -

This register allows the user to identify the number of bytes present in each TX and RX FIFO. The register is always

read zero when SERCOM_FIFO_IMPLEMENTED = 0. In this case, an access to this register will trigger a PAC slave

access error.

**Table 32-44.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | RXSPACE[4:0] | | | | |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

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

**Bits 12:8 – RXSPACE[4:0]** Rx FIFO Filled Space
These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

**Bits 4:0 – TXSPACE[4:0]** Tx FIFO Empty Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).
**Note:** When preloading is used, data written to shift register is not accessible in TXSPACE.


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

### 32.10.15. FIFO CPU Pointers

**Name:** FIFOPTR
**Offset:** 0x36
**Reset:** 0x0000
**Property:** -

This register provides a copy of internal CPU TX and RX FIFO pointers. The register is always read zero when SERCOM_FIFO_IMPLEMENTED = 0. In this case, an access to this register will trigger a PAC slave access error.

**Table 32-45.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPURDPTR[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPUWRPTR[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 11:8 – CPURDPTR[3:0]** RX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

**Bits 3:0 – CPUWRPTR[3:0]** TX FIFO Filled Space
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writting to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.


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

## 32.11. Register Summary - I2CM

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| | | 15:8 | | | | | SLEWRATE[1:0] | | FILTSEL[1:0] | |
| | | 23:16 | SEXTTOEN | MEXTTOEN | SDAHOLD[1:0] | | | | SMBUSEN | PINOUT |
| | | 31:24 | | LOWTOUTEN | INACTOUT[1:0] | | SCLSM | | SPEED[1:0] | |
| 0x04 | CTRLB | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | QCEN | SMEN |
| | | 23:16 | FIFOCLR[1:0] | | | | | ACKACT | CMD[1:0] | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TXTRHOLD[1:0] | | RXTRHOLD[1:0] | | FIFOEN | | | 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 | | | RXFF | TXFE | | SB | MB |
| 0x15 | Reserved | | | | | | | | | |
| 0x16 | INTENSET | 7:0 | ERROR | | | RXFF | TXFE | | SB | MB |
| 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTFLAG | 7:0 | ERROR | | | RXFF | TXFE | | 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 |
| 0x31 ... 0x33 | Reserved | | | | | | | | | |
| 0x34 | FIFOSPACE | 7:0 | | | | TXSPACE[4:0] | | | | |
| | | 15:8 | | | | RXSPACE[4:0] | | | | |
| 0x36 | FIFOPTR | 7:0 | | | | | CPUWRPTR[3:0] | | | |
| | | 15:8 | | | | | CPURDPTR[3:0] | | | |


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

### 32.11.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** RW

**Table 32-46.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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] | | | | SMBUSEN | PINOUT |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SLEWRATE[1:0] | | FILTSEL[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 arises 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 100 kHz baud rate.
These bits are not synchronized.


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

| Value | Name | Description |
|---|---|---|
| 0x0 | DIS | Disable |
| 0x1 | 55US | 5-6 SCL cycle time-out |
| 0x2 | 105US | 10-11 SCL cycle time-out |
| 0x3 | 205US | 20-21-6 SCL cycle time-out |

**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 "I$^2$C Host Behavioral Diagram (SCLSM = 0)" |
| 1 | SCL stretch only after ACK bit, "I$^2$C Host Behavioral Diagram (SCLSM = 1)". |

**Bits 25:24 – SPEED[1:0]** Transfer Speed

These bits define bus speed.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | SM | Standard-Mode (SM) and Fast-Mode (FM) |
| 0x1 | FMP | Fast-Mode Plus (FM+) |
| 0x2 | HS | High-Speed Mode |

**Bit 23 – SEXTTOEN** Slave SCL Low Extend Timeout

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 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 |

**Bit 22 – MEXTTOEN** Master SCL Low Extend Timeout

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 | Disable |
| 0x1 | 75NS | 50ns - 100ns hold time |
| 0x2 | 450NS | 300ns - 600ns hold time |
| 0x3 | 600NS | 400ns - 800ns hold time |


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

**Bit 17 – SMBUSEN** SMBUS Input Buffer Enable
This bit enables SMBus-compatible I/O logic level.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | SMBus input buffer is disabled. |
| 1 | SMBus input buffer is enabled. |

**Bit 16 – PINOUT** Pin Usage
This bit set 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. |

**Bits 11:10 – SLEWRATE[1:0]** Slew Rate Selection
This bit enables the I/O pins slew rate control.
This bit is not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | SMM | Standard mode |
| 0x1 | FM | Fast mode |
| 0x2 | FMP | Fast mode plus |
| 0x3 | HS | High-speed mode |

**Bits 9:8 – FILTSEL[1:0]** Input Filter Selection
These bits define filter length applied to the input signals.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIS | Disable |
| 0x2 | 50NS | Minimum 50ns filter |
| 0x3 | 10NS | Minimum 10ns filter |

**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.

| Value | Name | Description |
|---|---|---|
| 5 | I2C_MASTER | I2C host mode operation |

**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.


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

| 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 a bus 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.

**Notes:**

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by the 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.

## 32.11.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** RW

**Table 32-47.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FIFOCLR[1:0] | | | | | ACKACT | CMD[1:0] | |
| Access | R/W | R/W | | | | R/W | W | W |
| Reset | 0 | 0 | | | | 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 | | | | | | | | |

**Bits 23:22 – FIFOCLR[1:0]** FIFO Clear

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.SYSOP = 0.
These bits are not enable-protected.

| FIFOCLR[1:0] | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | TXFIFO | Clear TX FIFO |
| 0x2 | RXFIFO | Clear RX FIFO |
| 0x3 | BOTH | Clear both TX/RX FIFO |

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | TXFIFO | Clear TX FIFO |
| 0x2 | RXFIFO | Clear RX FIFO |
| 0x3 | BOTH | Clear both TX and RF FIFOs |

**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.


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

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).

**Table 32-48.** 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.

### 32.11.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** RW

**Table 32-49.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TXTRHOLD[1:0] | | RXTRHOLD[1:0] | | FIFOEN | | | DATA32B |
| 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 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

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

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

**Bits 31:30 – TXTRHOLD[1:0]** Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

| TXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated as long as the FIFO is not full. |
| 1 | HALF | Interrupt and DMA triggers are generated when half FIFO space is free. |
| 2 | EMPTY | Interrupt and DMA triggers are generated when the FIFO is empty. |
| 3 | - | Reserved |

**Bits 29:28 – RXTRHOLD[1:0]** Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

| RXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated when a DATA is present in the FIFO. |
| 1 | HALF | Interrupt and DMA triggers can be generated only when the FIFO is half-full. |
| 2 | FULL | Interrupt and DMA triggers can be generated only when the FIFO is full. |
| 3 | - | Reserved |

**Bit 27 – FIFOEN** FIFO Enable
This bit enables the FIFO operation.


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

| Value | Description |
|---|---|
| 0 | FIFO operation is disabled |
| 1 | FIFO operation is enabled |

**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.

### 32.11.4. Baud Rate

**Name:** BAUD
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** R/W

**Table 32-50.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Baud Rate Value 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 Baud Rate Value

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]** Baud Rate Value 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 "Clock Generation - Baud-Rate Generator".

**Bits 7:0 – BAUD[7:0]** Baud Rate Value

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.


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

For more information on how to calculate the frequency, see "Clock Generation - Baud-Rate Generator".


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

### 32.11.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00
**Property:** RW

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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

**Table 32-51.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | RXFF | TXFE | | SB | MB |
| Access | R/W | | | R/W | R/W | | R/W | R/W |
| Reset | 0 | | | 0 | 0 | | 0 | 0 |

**Bit 7 – ERROR** Combined Error Interrupt Disable
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.

**Bit 4 – RXFF** Rx FIFO Full Interrupt Disable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the RX FIFO Full bit, which disables the RX FIFO Full interrupt.

| Value | Description |
|---|---|
| 0 | The RX FIFO Full interrupt is disabled. |
| 1 | The RX FIFO Full interrupt is enabled. |

**Bit 3 – TXFE** Tx FIFO Empty Interrupt Disable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the TX FIFO Empty bit, which disables the TX FIFO Empty interrupt.

| Value | Description |
|---|---|
| 0 | The TX FIFO Empty interrupt is disabled. |
| 1 | The TX FIFO Empty interrupt is enabled. |

**Bit 1 – SB** Client On Bus Interrupt Disable
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 Disable
Writing '0' to this bit has no effect.


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

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.

### 32.11.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x16
**Reset:** 0x00
**Property:** RW

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). On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

**Table 32-52.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | RXFF | TXFE | | SB | MB |
| Access | R/W | | | R/W | R/W | | R/W | R/W |
| Reset | 0 | | | 0 | 0 | | 0 | 0 |

**Bit 7 – ERROR** Combined 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.

**Bit 4 – RXFF** Rx FIFO Full Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the RX FIFO Full bit, which enables the RX FIFO Full interrupt.

| Value | Description |
|---|---|
| 0 | The RX FIFO Full interrupt is disabled. |
| 1 | The RX FIFO Full interrupt is enabled. |

**Bit 3 – TXFE** Tx FIFO Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the TX FIFO Empty bit, which enables the TX FIFO Empty interrupt.

| Value | Description |
|---|---|
| 0 | The TX FIFO Empty interrupt is disabled. |
| 1 | The TX FIFO Empty 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.


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

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.

### 32.11.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** RW

**Note:** Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 32-53.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | RXFF | TXFE | | SB | MB |
| Access | R/W | | | R/W | R/W | | R/W | R/W |
| Reset | 0 | | | 0 | 0 | | 0 | 0 |

**Bit 7 – ERROR** Combined Error Interrupt
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: 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 4 – RXFF** Rx FIFO Full Interrupt
This flag is set when RX FIFO Threshold locations are fulfilled.
The flag is cleared when the RX FIFO is empty.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the RX FIFO Full interrupt flag.

**Bit 3 – TXFE** Tx FIFO Empty Interrupt
This flag is set when TX FIFO Threshold locations are available.
The flag is cleared when the TX FIFO is full.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the TX FIFO Empty interrupt flag.

**Bit 1 – SB** Client On Bus Interrupt
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 the SB bit 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.


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

**Bit 0 – MB** Host On Bus Interrupt

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.

### 32.11.8. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** RW

**Table 32-54.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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** 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.

| Value | Description |
|---|---|
| 0 | No Error |
| 1 | Transaction Length Error has occurred. |

**Bit 9 – SEXTTOUT** Client SCL Low Extend Timeout
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.

| Value | Description |
|---|---|
| 0 | No Time-Out |
| 1 | Client SCL Low Time-Out has occurred. |

**Bit 8 – MEXTTOUT** Host SCL Low Extend Timeout
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.


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

| Value | Description |
|---|---|
| 0 | No Time-Out |
| 1 | Host SCL Low Extend Time-Out has occurred |

**Bit 7 – CLKHOLD** Clock Hold

This bit is set when the host is holding the SCL line low, stretching the I²C 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.

| Value | Description |
|---|---|
| 0 | No Time-Out |
| 1 | SCL Low Time-Out has occurred |

**Bit 6 – LOWTOUT** SCL Low Timeout

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.

**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.
These bits are write synchronized.

| 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.


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

| Value | Description |
|---|---|
| 0 | No Error |
| 1 | Arbitration Lost |

**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 $I^2C$ 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 BUSERR.
If the $I^2C$ 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.

| Value | Description |
|---|---|
| 0 | No Error |
| 1 | Bus Error has occurred |


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

### 32.11.9. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** R

**Table 32-55.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.
Writing CTRLB.FIFOCLR when the SERCOM is enabled and Smart Mode 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. |


© 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. 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.

### 32.11.10. Address

**Name:** ADDR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** RW

**Table 32-56.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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]** 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 Mode
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** Length Enable


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

| Value | Description |
|---|---|
| 0 | Automatic transfer length disabled. |
| 1 | Automatic transfer length enabled. |

**Bits 10:0 – ADDR[10:0]** Address Value

When ADDR is written, the consecutive operation will depend on the bus state:

- **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.

**Note:** 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.

### 32.11.11. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** R/W

**Table 32-57.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Value

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 to this 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.

Writing or reading DATA.DATA when not in smart mode does not require synchronization.


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

## 32.11.12. Debug Control

**Name:** DBGCTRL
**Offset:** 0x30
**Reset:** 0x00
**Property:** R/W

**Table 32-58.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

**Bit 0 – DBGSTOP** Debug 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.

### 32.11.13. FIFO Space

**Name:** FIFOSPACE
**Offset:** 0x34
**Reset:** 0x0000
**Property:** R

**Table 32-59.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | RXSPACE[4:0] | | | | |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

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

**Bits 12:8 – RXSPACE[4:0]** Rx FIFO Filled Space
These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

**Bits 4:0 – TXSPACE[4:0]** Tx FIFO Empty Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).


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

### 32.11.14. FIFO CPU Pointers

**Name:** FIFOPTR
**Offset:** 0x36
**Reset:** 0x0000
**Property:** R/W

**Table 32-60.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPURDPTR[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPUWRPTR[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 11:8 – CPURDPTR[3:0]** CPU FIFO Read Pointer
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

**Bits 3:0 – CPUWRPTR[3:0]** CPU FIFO Write Pointer
These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.


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

## 32.12. Register Summary - I2CS

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | RUNSTDBY | | | MODE[2:0] | | | ENABLE | SWRST |
| | | 15:8 | | | | | SLEWRATE[1:0] | | FILTSEL[1:0] | |
| | | 23:16 | SEXTTOEN | | SDAHOLD[1:0] | | | | SMBUSEN | PINOUT |
| | | 31:24 | | LOWTOUT | | | SCLSM | | SPEED[1:0] | |
| 0x04 | CTRLB | 7:0 | | | | | | | | |
| | | 15:8 | AMODE[1:0] | | | | | AACKEN | GCMD | SMEN |
| | | 23:16 | FIFOCLR[1:0] | | | | | ACKACT | CMD[1:0] | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | | | | | SDASETUP[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | TXTRHOLD[1:0] | | RXTRHOLD[1:0] | | FIFOEN | | | DATA32B |
| 0x0C ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | INTENCLR | 7:0 | ERROR | | | RXFF | TXFE | DRDY | AMATCH | PREC |
| 0x15 | Reserved | | | | | | | | | |
| 0x16 | INTENSET | 7:0 | ERROR | | | RXFF | TXFE | DRDY | AMATCH | PREC |
| 0x17 | Reserved | | | | | | | | | |
| 0x18 | INTFLAG | 7:0 | ERROR | | | RXFF | TXFE | 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 | | SYSOP | 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] | | | | | | | |
| 0x2C ... 0x33 | Reserved | | | | | | | | | |
| 0x34 | FIFOSPACE | 7:0 | | | | TXSPACE[4:0] | | | | |
| | | 15:8 | | | | RXSPACE[4:0] | | | | |
| 0x36 | FIFOPTR | 7:0 | | | | | CPUWRPTR[3:0] | | | |
| | | 15:8 | | | | | CPURDPTR[3:0] | | | |


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

### 32.12.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** R/W

**Table 32-61.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | LOWTOUT | | | 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] | | | | SMBUSEN | 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SLEWRATE[1:0] | | FILTSEL[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 |
|---|---|---|---|---|---|---|---|---|
| | 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 – LOWTOUT** SCL Low Timeout Enable
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.

**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 $I^2C$ Behavioral Diagram (SCLSM = 0) |
| 1 | SCL stretch only after ACK bit according to $I^2C$ Client Behavioral Diagram SCLSM = 1 |

**Bits 25:24 – SPEED[1:0]** Transfer Speed
These bits define bus speed.
These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | SM | Standard-Mode (SM) and Fast-Mode (FM) |
| 0x1 | FMP | Fast-Mode Plus (FM+) |


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

| Value | Name | Description |
|---|---|---|
| 0x2 | HS | High-Speed Mode |

**Bit 23 – SEXTTOEN** Slave SCL Low Extend Timeout

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 | Disable |
| 0x1 | 75NS | 50ns - 100ns hold time |
| 0x2 | 450NS | 300ns - 600ns hold time |
| 0x3 | 600NS | 400ns - 800ns hold time |

**Bit 17 – SMBUSEN** SMBUS Input Buffer Enable

This bit enables SMBus-compatible I/O logic level.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | SMBus input buffer is disabled. |
| 1 | SMBus input buffer is enabled. |

**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 |

**Bits 11:10 – SLEWRATE[1:0]** Slew Rate Selection

This bit enables the I/O pins slew rate control.
These bits are not synchronized.
**Note:** If an $I^2C$ function is enabled on a pin, the corresponding PINCFGn.SLEWLIM for that pin MUST = 0x00.

| Value | Name | Description |
|---|---|---|
| 0x0 | SM | Standard mode |
| 0x1 | FM | Fast mode |
| 0x2 | FMP | Fast mode plus |
| 0x3 | HS | High-speed mode |

**Bits 9:8 – FILTSEL[1:0]** Input Filter Selection

These bits define filter length applied to the input signals.
These bits are not synchronized.


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

| Value | Name | Description |
|---|---|---|
| 0x0 | DIS | Disabled |
| 0x1 | 50F | Minimum 50ns filter (SCL fast mode) |
| 0x2 | 50E | Minimum 50ns filter (SDA even mode) |
| 0x3 | 10 | Minimum 10ns filter |

**Bit 7 – RUNSTDBY** Run during 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.

| Value | Name | Description |
|---|---|---|
| 0x4 | I2C_SLAVE | I2C client mode operation |

**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.
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 a bus 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.

**Notes:**

1. When the CTRLA.SWRST is written, the user should poll the SYNCB.SWRST bit to know when the reset operation is complete.
2. During a SWRST, access to registers/bits without SWRST are disallowed until the SYNCBUSY.SWRST is cleared by the 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.

### 32.12.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** RW

**Table 32-62.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FIFOCLR[1:0] | | | | | ACKACT | CMD[1:0] | |
| Access | R/W | R/W | | | | R/W | W | W |
| Reset | 0 | 0 | | | | 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 | | | | | | | | |

**Bits 23:22 – FIFOCLR[1:0]** FIFO Clear

When these bits are set, the corresponding FIFO will be cleared. The bits will automatically clear when SYNCBUSY.SYSOP = 0.
These bits are not enable-protected.

| FIFOCLR[1:0] | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | TXFIFO | Clear TX FIFO |
| 0x2 | RXFIFO | Clear RX FIFO |
| 0x3 | BOTH | Clear both TX/RX FIFO |

| Value | Name | Description |
|---|---|---|
| 0x0 | NONE | No action |
| 0x1 | TXFIFO | Clear TX FIFO |
| 0x2 | RXFIFO | Clear RX FIFO |
| 0x3 | BOTH | Clear both TX and RF FIFOs |

**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


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

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. This bit is not write-synchronized.
**Note:**
CTRLB.ACKACT shall not be updated more than once between each peripheral interrupt request.

| Value | Description |
|---|---|
| 0 | Send ACK |
| 1 | Send NACK |

**Bits 17:16 – CMD[1:0]** Command
This bit field triggers the client operation as shown 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. This bit is not write-synchronized.

**Table 32-63.** Command Description

| CMD[1:0] | STATUS.DIR Value | 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 |
| 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.
These bits are not write-synchronized.

| Value | Name | Description |
|---|---|---|
| 0x0 | MASK | The slave responds to the address written in ADDR.ADDR masked by the value in ADDR.ADDRMASK |
| 0x1 | 2ADDRS | The slave responds to the two unique addresses in ADDR and ADDRMASK |
| 0x2 | RANGE | The slave responds to the range of addresses between and including ADDR and ADDRMASK. ADDR is the upper limit |

**Bit 10 – AACKEN** Automatic Address Acknowledge
This bit enables the address to be automatically acknowledged if there is an address match.
This bit is not write-synchronized.

| 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.
This bit is not write-synchronized.


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

| 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. 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.

### 32.12.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** R/W

**Table 32-64.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TXTRHOLD[1:0] | | RXTRHOLD[1:0] | | FIFOEN | | | DATA32B |
| 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 |
|---|---|---|---|---|---|---|---|---|
| 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 |

**Bits 31:30 – TXTRHOLD[1:0]** Transmit FIFO Threshold
These bits define the threshold for generating the Data Register Empty interrupt and DMA TX trigger.

| TXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated as long as the FIFO is not full. |
| 1 | HALF | Interrupt and DMA triggers are generated when half FIFO space is free. |
| 2 | EMPTY | Interrupt and DMA triggers are generated when the FIFO is empty. |
| 3 | - | Reserved |

**Bits 29:28 – RXTRHOLD[1:0]** Receive FIFO Threshold
These bits define the threshold for generating the RX Complete interrupt and DMA RX trigger.

| RXTRHOLD | Name | Description |
|---|---|---|
| 0 | DEFAULT | Interrupt and DMA triggers can be generated when a DATA is present in the FIFO. |
| 1 | HALF | Interrupt and DMA triggers can be generated only when the FIFO is half-full. |
| 2 | FULL | Interrupt and DMA triggers can be generated only when the FIFO is full. |
| 3 | - | Reserved |

**Bit 27 – FIFOEN** FIFO Enable
This bit enables the FIFO operation.


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

| Value | Description |
|---|---|
| 0 | FIFO operation is disabled |
| 1 | FIFO operation is enabled |

**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:

$t_{SU:DAT} = (\text{CLK_SERCOMx} \times \text{APB period}) \times (6 + 16 \times \text{SDASETUP})$


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

### 32.12.4. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x14
**Reset:** 0x00
**Property:** RW

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) On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

**Table 32-65.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | RXFF | TXFE | DRDY | AMATCH | PREC |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Combined Error Interrupt Disable
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.

**Bit 4 – RXFF** RX FIFO Full Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the RX FIFO Full bit, which disables the RX FIFO Full interrupt.

| Value | Description |
|---|---|
| 0 | The RX FIFO Full interrupt is disabled. |
| 1 | The RX FIFO Full interrupt is enabled. |

**Bit 3 – TXFE** TX FIFO Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the TX FIFO Empty bit, which disables the TX FIFO Empty interrupt.

| Value | Description |
|---|---|
| 0 | The TX FIFO Empty interrupt is disabled. |
| 1 | The TX FIFO Empty interrupt is enabled. |

**Bit 2 – DRDY** Data Interrupt Disable
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.

**Bit 1 – AMATCH** Address Match Interrupt Disable
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.

**Bit 0 – PREC** Stop Received Interrupt Disable
Writing '0' to this bit has no effect.


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

Writing '1' to this bit will clear the Stop Received Interrupt Enable bit, which disables the Stop Received interrupt.


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

### 32.12.5. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x16
**Reset:** 0x00
**Property:** RW

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) On read, a bit value of zero indicates the associated interrupt is disabled while a bit value of one indicates the associated interrupt is enabled.

**Table 32-66.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | RXFF | TXFE | DRDY | AMATCH | PREC |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Combined 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 4 – RXFF** Rx FIFO Full Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the RX FIFO Full bit, which enables the RX FIFO Full interrupt.

| Value | Description |
|---|---|
| 0 | The RX FIFO Full interrupt is disabled. |
| 1 | The RX FIFO Full interrupt is enabled. |

**Bit 3 – TXFE** Tx FIFO Empty Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will set the TX FIFO Empty bit, which enables the TX FIFO Empty interrupt.

| Value | Description |
|---|---|
| 0 | The TX FIFO Empty interrupt is disabled. |
| 1 | The TX FIFO Empty interrupt is enabled. |

**Bit 2 – DRDY** Data 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. |


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

**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.

### 32.12.6. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00
**Property:** RW

**Note:** Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

**Table 32-67.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | ERROR | | | RXFF | TXFE | DRDY | AMATCH | PREC |
| Access | R/W | | | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bit 7 – ERROR** Combined Error Interrupt
This bit is set when any error is detected. Errors that will set this flag have corresponding status flags in the STATUS register: LENERR, SEXTTOUT, LOWTOUT, COLL, and BUSERR.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the flag.

**Bit 4 – RXFF** RX FIFO Full
This flag is set when RX FIFO Threshold locations are fulfilled.
The flag is cleared when the RX FIFO is empty.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the RX FIFO Full interrupt flag.

**Bit 3 – TXFE** TX FIFO Empty
This flag is set when TX FIFO Threshold locations are available.
The flag is cleared when the TX FIFO is full.
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the TX FIFO Empty interrupt flag.

**Bit 2 – DRDY** Data Interrupt
This flag is set when a $I^2C$ client byte transmission is successfully completed.
The flag is cleared by hardware by 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 Interrupt
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.


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

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 Interrupt

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.

### 32.12.7. Status

**Name:** STATUS
**Offset:** 0x1A
**Reset:** 0x0000
**Property:** R/W

**Table 32-68.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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** Slave SCL Low Extend Timeout
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.


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

**Bit 6 – LOWTOUT** SCL Low Timeout

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 0x3 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 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 immediately released the SDA and SCL lines and waited 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 this bit.


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

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. |
| 1 | Bus error detected. |


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

## 32.12.8. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** R

**Table 32-69.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | SYSOP | ENABLE | SWRST |
| Access | | | | R | | R | R | R |
| Reset | | | | 0 | | 0 | 0 | 0 |

**Bit 4 – LENGTH** Length Synchronization Busy

Writing the LENGTH register requires synchronization. When written, this bit will be set until synchronization is complete. If LENGTH is written while SYNCBUSY.LENGTH is asserted, a bus 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 2 – SYSOP** System Operation Synchronization Busy

Writing CTRLB.FIFOCLR when the SERCOM is enabled requires synchronization. In this case 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. |


© 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.

### 32.12.9. Length

**Name:** LENGTH
**Offset:** 0x22
**Reset:** 0x0000
**Property:** R/W

**Table 32-70.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 32.12.10. Address

**Name:** ADDR
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** R/W

**Table 32-71.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Value
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).


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

| 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.

### 32.12.11. Data

**Name:** DATA
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** R/W

**Table 32-72.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Value

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.

### 32.12.12. FIFO Space

**Name:** FIFOSPACE
**Offset:** 0x34
**Reset:** 0x0000
**Property:** R

**Table 32-73.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | RXSPACE[4:0] | | | | |
| Access | | | | R | R | R | R | R |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

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

**Bits 12:8 – RXSPACE[4:0]** Rx FIFO Filled Space
These bits return the number filled locations in the RX FIFO (bytes or words, depending on CTRLC.DATA32B setting).

**Bits 4:0 – TXSPACE[4:0]** Tx FIFO Empty Space
These bits return the number of available locations in the TX FIFO (bytes or words, depending on CTRLC.DATA32B setting).


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

### 32.12.13. FIFO CPU Pointers

**Name:** FIFOPTR
**Offset:** 0x36
**Reset:** 0x0000
**Property:** R/W

**Table 32-74.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPURDPTR[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | CPUWRPTR[3:0] | | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 11:8 – CPURDPTR[3:0]** CPU FIFO Read Pointer
These bits return the CPURDPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. Reading DATA register, will return RXFIFO[CPURDPTR] location value.

**Bits 3:0 – CPUWRPTR[3:0]** CPU FIFO Write Pointer
These bits return the CPUWRPTR pointer value. These bits can be written only if the SERCOM is halted during debugging. When writing to DATA register, the DATA will be written to TXFIFO[CPUWRPTR] location.


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

# 33. Controller Area Network (CAN)

## 33.1. Overview

The CAN communication controller provides hardware support for the CAN protocol version 2.0 part A and B plus ISO 11898-1:2015 which includes the CAN Flexible Data Rate Frame format (CAN-FD).

For FIFO and filter storage, the CAN module uses system RAM, which it accesses through a Host AHB bus. Each CAN FD instance may use up to 4864 bytes of system memory.

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.

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.

## 33.2. Features

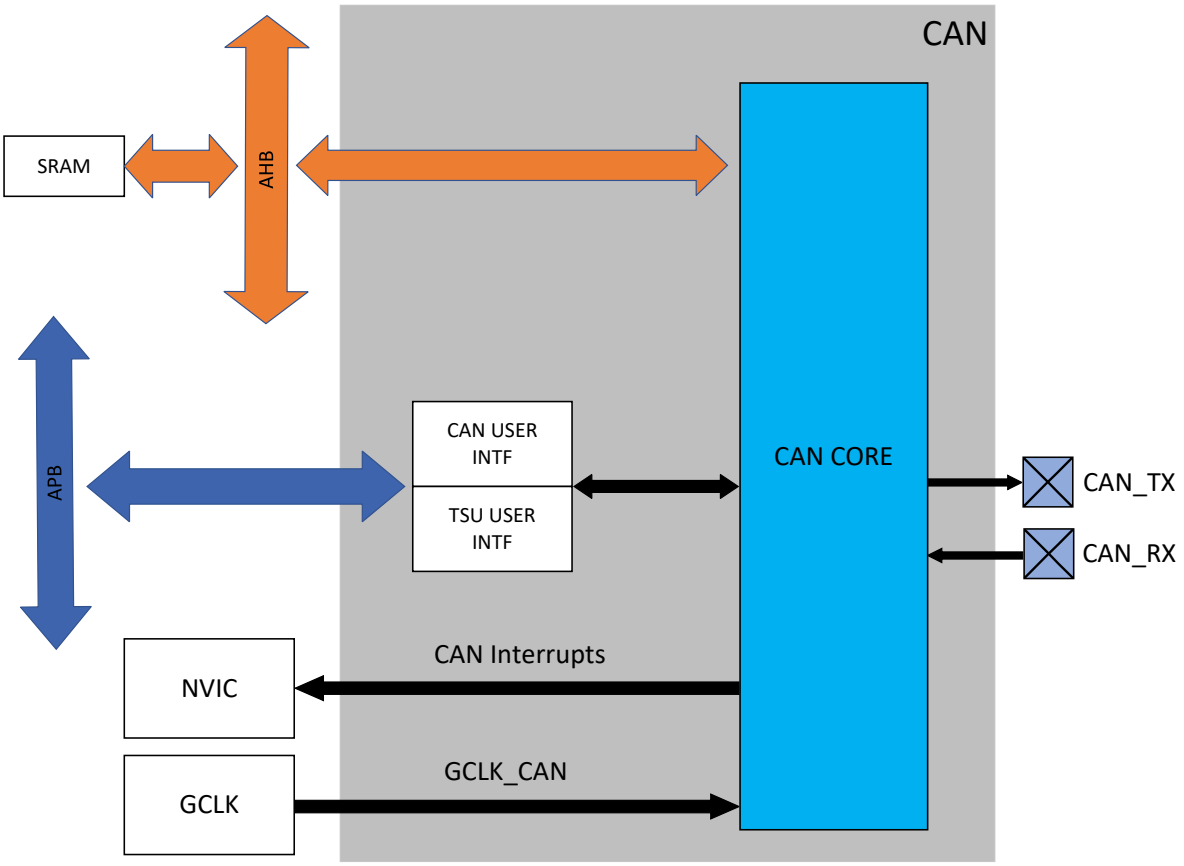
The following are key features of the CAN module:

- Conforms with CAN protocol version 2.0 part A and B and ISO 11898-1:2015
- Up to two Controller Area Network (CAN)
  - Supporting CAN 2.0 A/B and CAN-FD (ISO 11898-1:2015)
- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- Acceptance filtering
  - Up to 128 configurable filter elements
  - Special filter for SAE J1939 IDs
  - Separate signaling on reception of High Priority Messages
- 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
- CiA 603 hardware timestamping support
- External Timestamping Unit (TSU)


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

## 33.3. Block Diagram

**Figure 33-1.** Block Diagram




## 33.4. Signal Description

**Table 33-1.** Signal Description

| PIN NAME | DESCRIPTION | TYPE |
|---|---|---|
| CAN0_RX | CAN0_FD Receive | Digital Input |
| CAN0_TX | CAN0_FD Transmit | Digital Output |
| CAN1_RX | CAN1_FD Receive | Digital Input |
| CAN1_TX | CAN1_FD Transmit | Digital Output |

## 33.5. Functional Description

### 33.5.1. Operating Modes

#### 33.5.1.1. 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


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

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 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’

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’.

**Note:** It is recommended to initialize the Message RAM after hardware reset by writing e.g. 0x00000000 to each Message RAM word to create valid parity/ECC checksums. This avoids that reading from uninitialized Message RAM sections will activate interrupt IR.BEC (Bit Error Corrected) or IR.BEU (Bit Error Uncorrected).

#### 33.5.1.2. Normal Operation

Once the CAN is initialized and the CCCR.INIT bit (CCCR <0>) 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 the 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.

#### 33.5.1.3. 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 standard CAN frames with 11-bit identifiers and the first previously reserved bit in extended CAN frames with 29-bit identifiers will now be decoded as FDF bit. A recessive FDF signifies a CAN FD frame, where as a dominant FDF bit signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF (i.e., 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 a dominant res bit and a recessive BRS bit. The coding of a recessive res bit is reserved for future expansion of the protocol. In case the CAN receives a frame with a recessive FDF bit and a recessive res bit, it will signal a Protocol Exception Event by setting the PSR.PXE bit (PSR <14>). When Protocol Exception Handling is enabled (CCCR.PXHD bit (CCCR<12>) = ‘0’), this causes the operation state to change from Receiver (PSR.ACT = “0x2”) to synchronizing on CAN communication (PSR.ACT bits (PSR <4:3>) = “0x0”) at the next sample point. In case Protocol Exception Handling is disabled (CCCR.PXHD bit (CCCR<12>) = ‘1’), the CAN will treat a recessive res bit as a form error and will respond with an error frame.

The CAN FD operation is enabled by programming the CCCR.FDOE bit (CCCR<8>). In case the CCCR.FDOE bit (CCCR<8>) = ‘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, it can be configured through the FDF bit in the respective Tx Buffer element. With the CCCR.FDOE bit (CCCR<8>) = ‘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 the FDF bit of a Tx


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

Buffer element is set. The CCCR.FDOE bit (CCCR<8>) and the CCCR.BRSE bit (CCCR<9>) can only be changed while the CCCR.INIT bit (CCCR <0>) and the CCCR.CCE bit (CCCR <1>) are both set.

With the CCCR.FDOE bit (CCCR<8>) = '0', the setting of FDF and BRS bits is ignored and frames are transmitted in Classic CAN format. With the CCCR.FDOE bit (CCCR<8>) = '1' and the CCCR.BRSE bit (CCCR<9>) = '0', only the FDF bit of a Tx Buffer element is evaluated. With the CCCR.FDOE bit (CCCR<8>) = '1' and the CCCR.BRSE bit (CCCR<9>) = '1', transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with FDF and BRS bits 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. However, the codes 9 to 15, (which in standard CAN all code a data field of 8 bytes), are coded according to the following table.

**Table 33-2.** 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 and Prescaler Register (NBTP). In the following CAN FD data phase, the fast CAN bit timing is used as defined by the Data Bit Timing and Prescaler Register (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_CANx), for example, with a CAN clock frequency of 20 MHz and the shortest configurable bit time of 4 $t_q$, the bit rate in the data phase is 5 Mbits.

In both data frame formats, CAN FD long and CAN FD fast, the value of the Error Status Indicator (ESI) bit 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.

#### 33.5.1.4. 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 CANx_TX the CAN receives the transmitted data from its local CAN transceiver via pin CANx_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.

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.


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

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 the DBTP.TDC bit (DBTP<23>).

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 CANx_TX through the transceiver to the receive input CANx_RX plus the transmitter delay compensation offset as configured by TDCR.TDCO bits (TDCR <15:8>). 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 minimum time quantum (mtq). 1 mtq is equal to time period of GCLK_CANx clock.

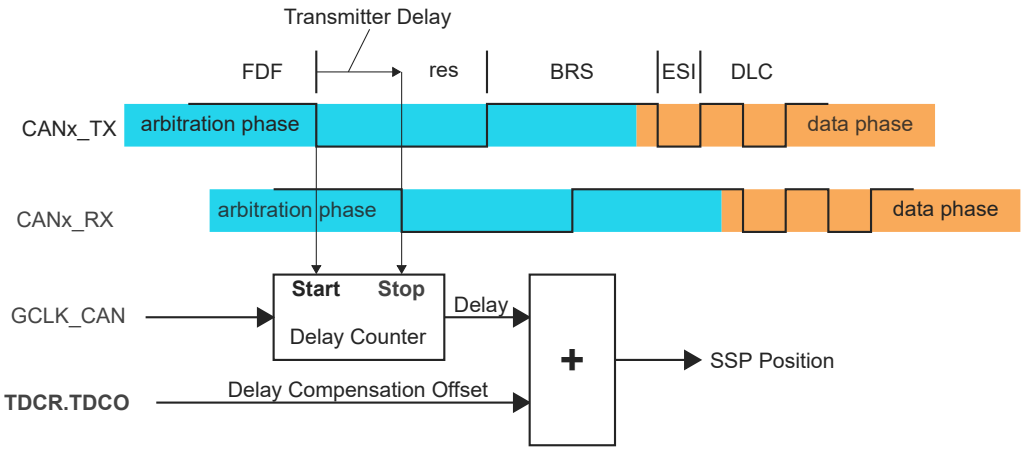
The PSR.TDCV bits (PSR <22:16>) show the actual transmitter delay compensation value. PSR.TDCV bits (PSR <22:16>) are cleared when CCCR.INIT bit (CCCR <0>) is set and is updated at each transmission of an FD frame while DBTP.TDC bit (DBTP <23>) is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the CAN:

- The sum of the measured delay from CANx_TX to CANx_RX and the configured transceiver delay compensation offset FBTP.TDCR bits (TDCR <15:8>) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from CANx_TX to CANx_RX and the configured transceiver delay compensation offset FBTP.TDCR bits (TDCR <15:8>) 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 the DBTP.TDC bit (DBTP <23>) = '1', the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input CANx_TX of the transmitter. The resolution of this measurement is one mtq.

**Figure 33-2.** Transceiver delay measurement





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

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 bit (TDCR <6:0>). This defines a minimum value for the SSP position. Dominant edges of CANx_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 bit (TDCR <6:0>) and CANx_RX is low.

#### 33.5.1.5. 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 (the ECR.REC bit (ECR <14:8>), the ECR.TEC bit (ECR <7:0>)) are frozen while Error Logging (the ECR.CEL bit (ECR <23:16>)) is still incremented. The CPU can set the CAN into Restricted Operation mode by setting the CCCR.ASM bit (CCCR <2>). The bit can only be set by the CPU when both of the CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) 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 the CCCR.ASM bit (CCCR <2>).

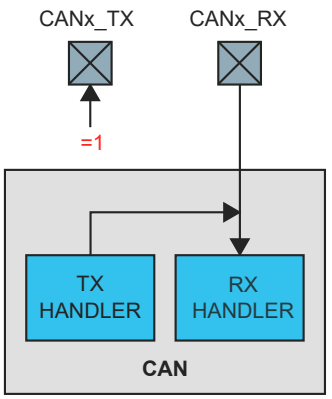
The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

#### 33.5.1.6. Bus Monitoring Mode

The CAN is set in Bus Monitoring Mode by programming the CCCR.MON bit (CCCR <5>) 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 (the 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 the 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, CANx_TX and CANx_RX, to the CAN in Bus Monitoring Mode.

**Figure 33-3.** Pin Control in Bus Monitoring Mode




**Bus Monitoring Mode**


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

#### 33.5.1.7. 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 through the CCCR.DAR bit (CCCR <6>).

**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:
  - The Corresponding Tx Buffer Transmission Occurred bit (TXBTO.TOx) is set
  - The Corresponding Tx Buffer Cancellation Finished bit (TXBCF.CFx) is not set
- Successful transmission in spite of cancellation:
  - The Corresponding Tx Buffer Transmission Occurred bit (TXBTO.TOx) is set
  - The Corresponding Tx Buffer Cancellation Finished bit (TXBCF.CFx) is set
- Arbitration lost or frame transmission disturbed:
  - The Corresponding Tx Buffer Transmission Occurred bit (TXBTO.TOx) is not set
  - The Corresponding Tx Buffer Cancellation Finished bit (TXBCF.CFx) is 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).

#### 33.5.1.8. Test Modes

To enable write access to the TEST register, the CCCR.TEST bit (CCCR <7>) must 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 the TEST.TX bits (TEST <6:5>). 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 the TEST.RX bit (TEST <7>). Both functions can be used to check the CAN bus physical layer.

Due to the synchronization mechanism between the GCLK_CANx and CLK_CANx_AHB domains, there may be a delay of several CLK_CANx_AHB periods between writing to the TEST.TX bits (TEST <6:5>) until the new configuration is visible at the output pin, CANx_TX. This also applies when reading the input pin, CANx_RX, through the TEST.RX bit (TEST <7>).

**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.

##### 33.5.1.8.1. External Loop Back Mode

The CAN can be set in External Loop Back Mode by programming the TEST.LBCK bit (TEST <4>) 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 signals CANx_TX and CANx_RX to the CAN in External Loop Back Mode.

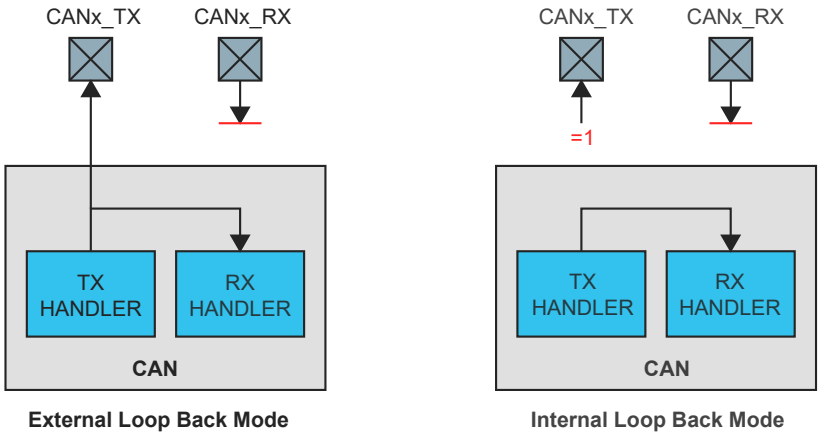
This mode is provided for hardware self-test. To be independent from external stimulation, the CAN ignores acknowledge errors (the recessive bit sampled in the acknowledge slot of a data/remote frame) in Loop Back Mode. In this mode the CAN performs an internal feedback from its Tx output to its Rx input. 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.


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

#### 33.5.1.8.2. Internal Loop Back Mode

Internal Loop Back Mode is entered by programming the TEST.LBCK bit (TEST <4>) and the CCCR.MON bit (CCCR <5>) 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 following figure shows the connection of the CANx_TX and CANx_RX pins to the CAN in case of Internal Loop Back Mode.

**Figure 33-4.** Pin Control in Loop Back Modes




### 33.5.1.9. 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.

#### 33.5.1.9.1. 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 or disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are as follows:

- 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:


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

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag, IR.HPM (IR <8>)
- Set High Priority Message interrupt flag, IR.HPM (IR <8>), 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 the PSR.LEC bits (PSR <2:0>) respectively the 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 bits (PSR <2:0>) respectively PSR.FLEC bit. In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in *Rx FIFO Overwrite Mode* have to 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.

In case 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.

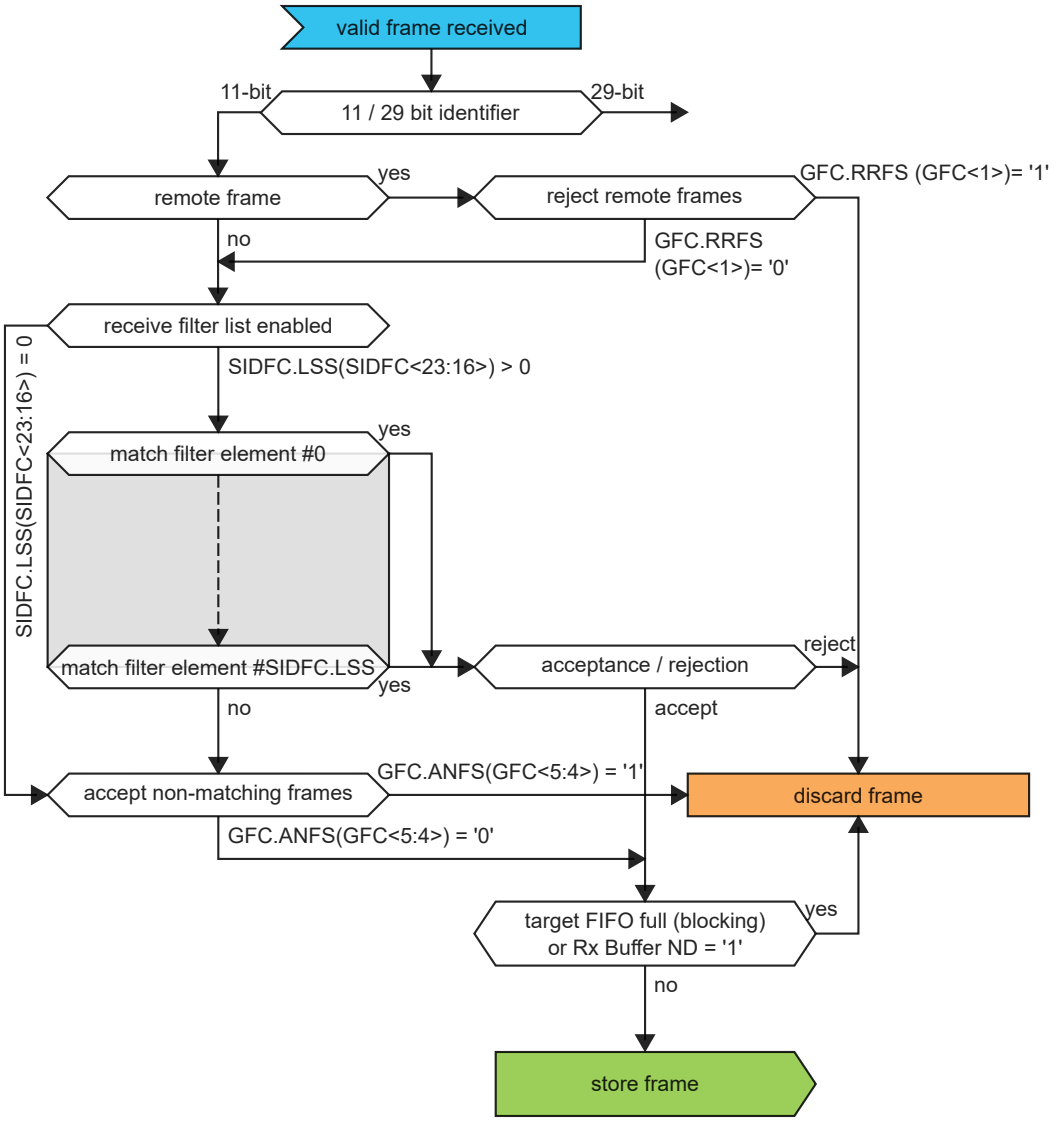
**Standard Message ID Filtering**


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

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*.

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC Message ID, the Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

**Figure 33-5.** Standard Message ID Filtering




### Extended Message ID Filtering

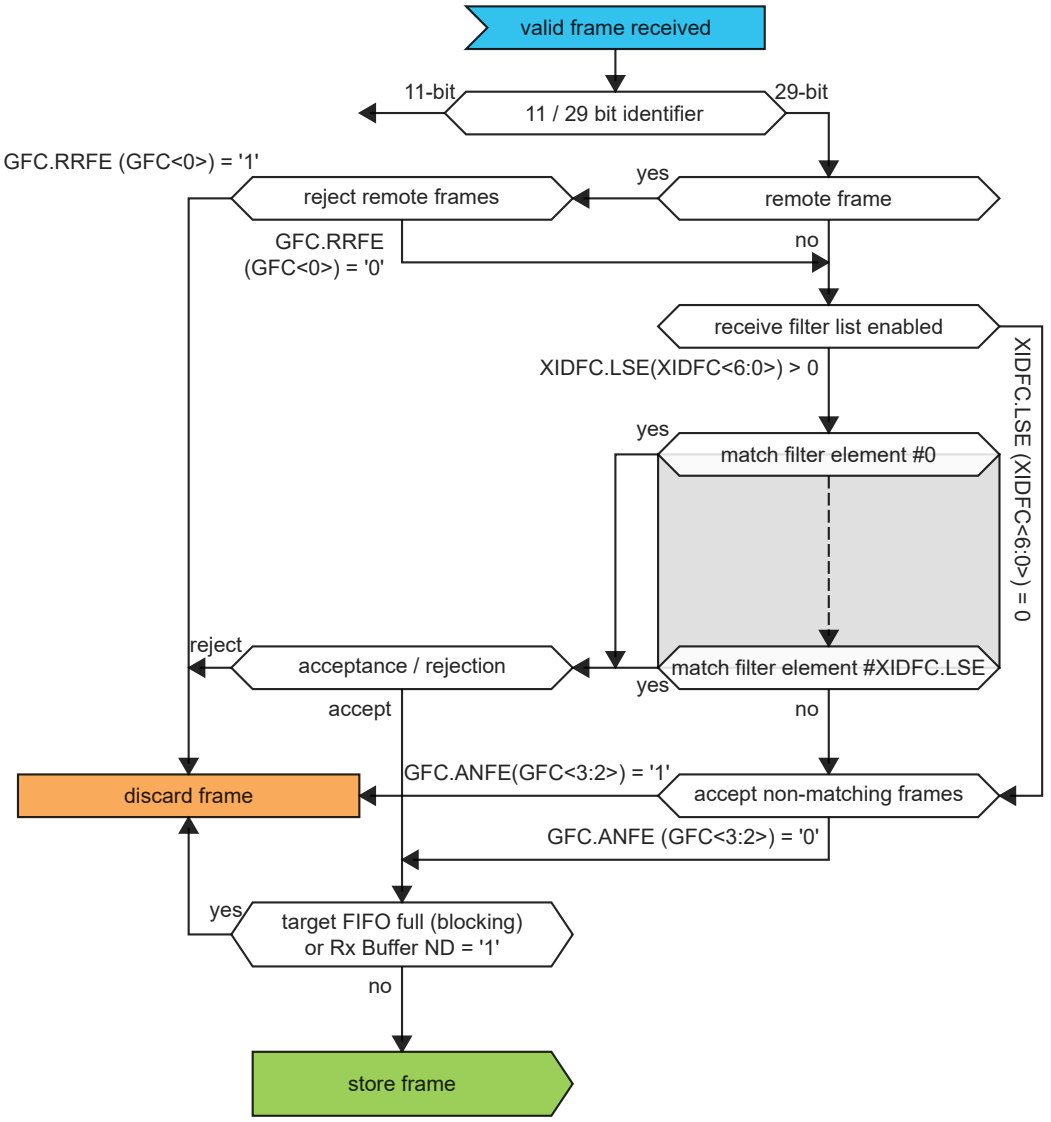
The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The extended Message ID Filter element is described in *Extended Message ID Filter Element*.

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 33-6.** Extended Message ID Filtering




### 33.5.1.9.2. Rx FIFOs

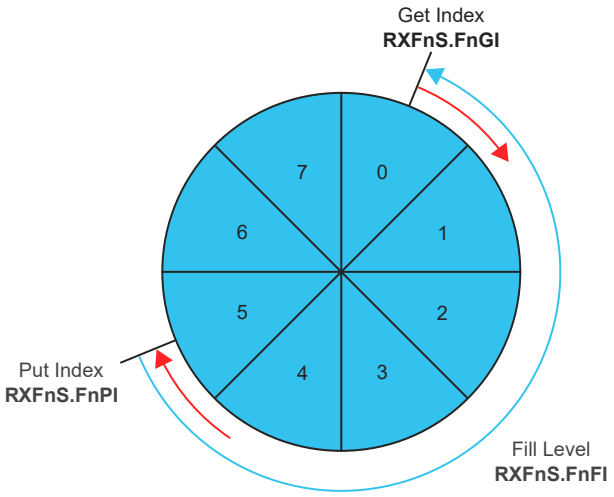
The Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done through the RXF0C and RXF1C registers.

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, where n = 0 for Rx FIFO 0 buffer and n = 1 for Rx FIFO 1 buffer.


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

**Figure 33-7.** Rx FIFO Status (n = 0 for Rx FIFO 0 buffer and n = 1 for Rx FIFO 1 buffer)




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 33-3.** 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.

If 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.

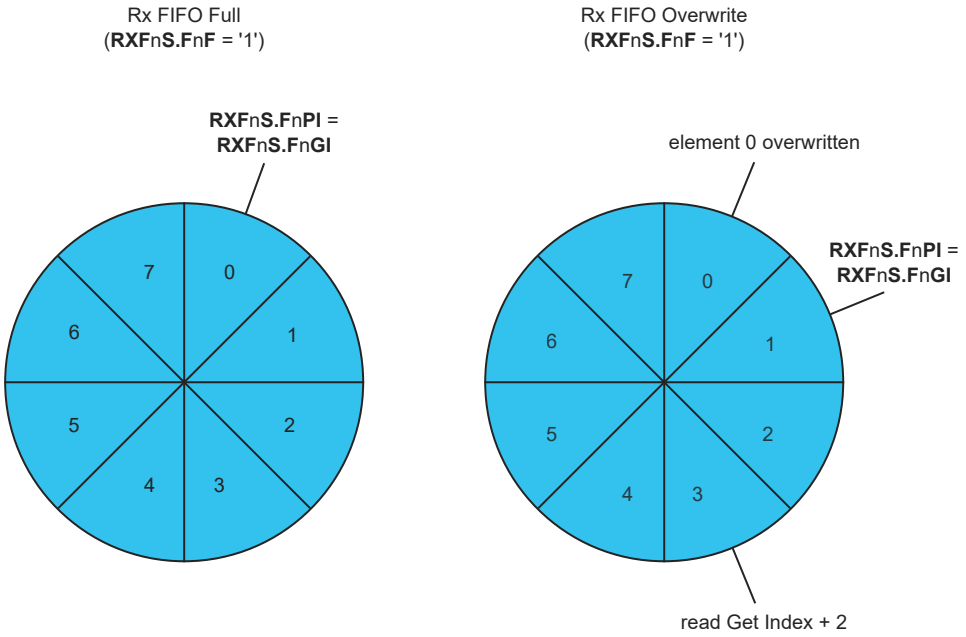
When an Rx FIFO full condition (RXFnS.FnPI = RXFnS.FnGI) is signaled by RXFnS.FnF = 1, the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

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 33-8.** Rx FIFO Overflow Handling (n = 0 for Rx FIFO 0 buffer and n = 1 for Rx FIFO 1 buffer)




After reading from the Rx FIFO, the number of the last element read must 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).

### 33.5.1.9.3. Dedicated Rx Buffers

The CAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured through the RXBC.RBSA bits (RXBC <15:0>).

For each Rx Buffer a Standard or Extended Message ID Filter Element with SFEC/EFEC = 111 and SFID2 / EFID2[10:9] = 00 must 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 bit (IR <19>) (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

**Table 33-4.** 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 the NDAT1, NDAT2 registers are set. As long as the New Data flag is set,


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

the 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, the IR.DRX bit (IR <19>)
- The Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

### 33.5.1.9.4. Debug on CAN Support

Debug messages are stored into Rx Buffers. For debug handling, three consecutive Rx Buffers, such as #61, #62, #63 must 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 the RXBC.RBSA bits (RXBC <15:0>)), no additional configuration is required.

For filtering of debug messages Standard or Extended Filter Elements with SFEC/EFEC = 111 must 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 must be programmed to '111'. In this case the SFID1 / SFID2 and EFID1 / EFID2 fields 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 the IR.DRX bit (IR <19>) are set. The reception of debug messages can be monitored through the RXF1S.DMS bits (RXF1S <31:30>).

**Table 33-5.** 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 |

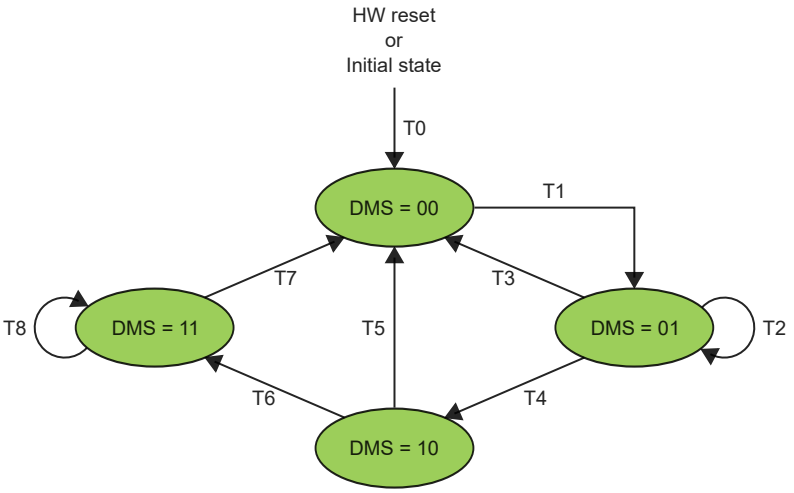
**Debug Message Handling**

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


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

DMA request is activated only when all three debug messages A, B, and C have been received in correct order.

**Figure 33-9.** 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)

### 33.5.1.10. 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 33-6.** 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.


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

#### 33.5.1.10.1. 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 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.

For example, if 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 the CCCR.TXP bit (CCCR<14>). 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 bit (CCCR<14>) = 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.

#### 33.5.1.10.2. 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 through 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 eddicated 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, the TXBC.TBSA bit (TXBC <15:0>).

**Table 33-7.** 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 |

#### 33.5.1.10.3. Tx FIFO

Tx FIFO operation is configured by programming the TXBC.TFQM bit (TXBC <30>) to '0'. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index TXFQS.TFGI bit TXFQS (<12:8>). After each transmission the Get Index is incremented cyclically until


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

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 bits (TXFQS <5:0>) as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages must be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index TXFQS.TFQPI bits TXFQS (<20:16>). An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, the Tx FIFO Full (TXFQS.TFQF bit ( TXFQS <21>) = '1') is signaled. In 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 through 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 *Tx Buffer / FIFO / Queue Element Size*). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI bits (TXFQS <20:16>) (0...31). Element Size to the Tx Buffer Start Address TXBC.TBSA bits ( TXBC <15:0>).

#### 33.5.1.10.4. Tx Queue

The Tx Queue operation is configured by programming the TXBC.TFQM bit (TXBC <30>) 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 bits TXFQS (<20:16>). An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS.TFQF bit (TXFQS <21>) = 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 the TXBRP register instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

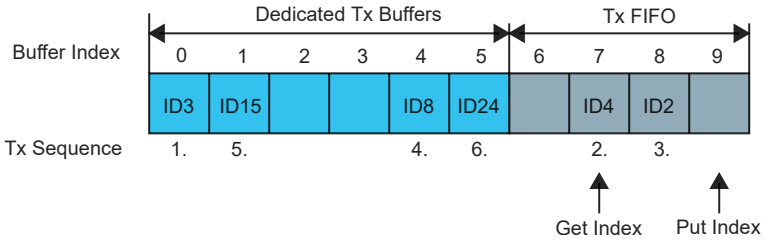
A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (refer to the *Tx Buffer / FIFO / Queue Element Size*). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI bits TXFQS (<20:16>) (0...31). Element Size to the Tx Buffer Start Address TXBC.TBSA bits ( TXBC <15:0>).

#### 33.5.1.10.5. 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 the TXBC.NDTB bits (TXBC <21:16>). The number of Tx Buffers assigned to the Tx FIFO is configured by the TXBC.TFQS bits (TXBC <29:24>). In case the TXBC.TFQS bits (TXBC <29:24>) is programmed to zero, only Dedicated Tx Buffers are used.


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

**Figure 33-10.** Example of mixed Configuration Dedicated Tx Buffers/Tx FIFO



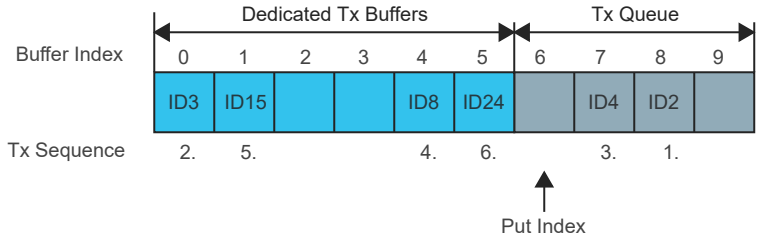

Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TXFQS.TFGI bits (TXFQS <12:8>)).
- Buffer with lowest Message ID gets highest priority and is transmitted next.

### 33.5.1.10.6. 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 the TXBC.NDTB bits (TXBC <21:16>). The number of Tx Queue Buffers is configured by the TXBC.TFQS bits (TXBC <29:24>). In case the TXBC.TFQS bits (TXBC <29:24>) is programmed to zero, only Dedicated Tx Buffers are used.

**Figure 33-11.** 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.

### 33.5.1.10.7. 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 (eqal to number of Tx Buffer) of the TXBCR register. Transmit cancellation is not intended for the Tx FIFO operation.

Successful cancellation is signaled by setting the corresponding bit of the TXBCF register 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.


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

#### 33.5.1.10.8. 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 the IR.TEFF bit (IR <14>), 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 the Interrupt Flag IR.TEFL bit (IR <15>) 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 the TXEFC.EFWM bit (TXEFC <29:24>), the Interrupt Flag IR.TEFW bit (IR <13>) is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index TXEFS.EFGI bit (TXEFS <12:8>) has to be added to the Tx Event FIFO start address TXEFC.EFSA bit (TXEFC <15:0>).

#### 33.5.1.11. 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 the *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.

### 33.5.2. Timestamp Generation

#### 33.5.2.1. Default Behavior

For internal 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 via TSCV.TSC. A write access to register TSCV resets the counter to zero. 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.

By programming bit TSCC.TSS, 16-bit external counter can be used, instead of the internal 16-bit counter. The Counter value can be provided by the TSU extension.

#### 33.5.2.2. Behavior when Enabling the TSU

**Overview**


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For the AUTOSAR (Automotive Open System Architecture) compatible synchronization of time bases between CAN nodes, only software implementations had been used due to a lack of special hardware. To further increase the time accuracy, special hardware is required. CiA 603 specifies a hardware-based concept, which is implemented into the TSU. This approach is independent of interrupt response times and thus achieves the best possible accuracy. The TSU may operate on its own internal timebase, or uses an external time base, e.g. a reference time base from a timer/counter within the SoC.

The TSU offers the capability to store up to 8 captured timestamps. The resolution depends on the internal or external counter resolution. Up to 32-bit resolution is available. To enable the CAN-FD hardware timestamping together with the TSU, the CCCR.UTSU bit must be set to 1. If the bit is cleared, the internal 16-bit timestamp counter will be used by default. A message timestamping can happen at the start-of-frame (SoF), or end-of-frame (EoF).

The TSU is configured via the TSCFG register, which is writable only when the CCCR.CCE = 1. With TSCFG.TSUE the TSU is enabled. Default after reset is disabled. The Timebase Prescaler TSCFG.TBPRE is used to adapt the count rate of the TSU's internal counter, if enabled, to the requirements of the application. Either the internal timebase or an external timebase input can be selected for timestamping via TSCFG.TBCS. In case TSCFG.TBCS = '0', the internal free running timebase counter is used. In case TSCFG.TBCS = '1', the timestamp value is taken from the Timebase Input tsu_tbin[31:0]. When the internal timebase has been excluded during synthesis (CAN_INTERNAL_TB_SOF parameter is set to 0 or 2), there is no internal timebase counter available and TSCFG.TBCS is fixed to '1'.

**Reception of Timestamped Messages**

To configure the M_CAN for reception of timestamped messages, a Standard/Extended Message ID Filter Element has to be set up by configuration of S0.SSYNC = '1' respectively F1.ESYNC = '1'. In case the filter element matches the received frame and the received frame is valid, a capture will be performed at the EOF. The value of the Actual Timebase (ATB) is stored into the TSU's timestamp register selected by the TSS2.TSP. The TSP value from the TSU will be written into the R1.RXTSP of the related RX Buffer or RX FIFO element.

**Transmission of Timestamped Messages**

In case bits T1.TSCE and T1.EFC of a TX Buffer element are set, a successful transmission of that TX Buffer triggers the storage of the value of the TSU's Actual Timebase ATB to the TSU's timestamp register selected by the Timestamp Pointer TSS2.TSP. The Timestamp Pointer value from the TSU is written to the E1.TXTSP of the related TX Event FIFO element.

**Capture Point**

TSCFG.SCP configures the point in time when a timestamp is captured for Sync messages (0: at EOF, 1: at SOF of the received/transmitted Sync message).

When TSCFG.SCP = '0', the timestamp is captured directly from the Actual Timebase ATB to the timestamp register TSn, when the message gets valid at EOF. This configuration is intended for AUTOSAR conformant applications.

When TSCFG.SCP = '1', the Actual Timebase ATB is captured into a temporary buffer at each SOF. When the message gets valid at EOF the timestamp is copied from the temporary buffer to the timestamp register TSn.

**Handling of Timestamps**

Depending on the device configuration, the TSU stores up to sixteen 32-bit timestamps. Refer to the device DOS for details on the effective number of timestamps.

The timestamps are written to TS0 to TSn in a cyclic manner. Each timestamp register has assigned two status bits:

- TSS1.TSNn is set whenever a timestamp was stored to the related timestamp register TSn.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- TSS1.TSLn is set when a new timestamp was stored to the related timestamp register TSn before the previously store timestamp has been read out.

Reading a Timestamp register resets the related TSS1 bits.

**Synchronization process after AUTOSAR**

After successful configuration of all participating CAN nodes, the timers of the time slaves can be corrected with a two-step synchronization process. In the first step, the current time T0 is latched in the time master, and the part of T0, which represents the seconds, is sent to the time slaves with the SYNC message. If the EoF (End-Of-Frame) is reached when sending this SYNC message, then timestamps are stored in the TSUs of all nodes, i.e. in the Time Master and all Time Slaves. This is the common point of time reference. In the second step, the time master adds its TSU time stamp to the nanosecond part of T0 and sends it to the time slaves via FollowUp message. From this, the time slaves can derive their own time base.

**Saving time stamps at SoF (Start-of-Frame)**

For non-AUTOSAR applications, the TSU can also be configured to store timestamps at the start of a CAN message (SoF bit).

### 33.5.3. Timeout Counter

To signal 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 the TSCC.TCP bit (TSCC <19:16>) as the Timestamp Counter. The Timeout Counter is configured through the TOCC register. The actual counter value can be read from the TOCV.TOC bits (TOCV <15:0>). The Timeout Counter can only be started while the CCCR.INIT bit (CCCR <0>) = 0. It is stopped when the CCCR.INIT bit (CCCR <0>) = 1, for example, when the CAN enters 'bus off' state.

The operation mode is selected by the TOCC.TOS bits (TOCC <2:1>). When operating in continuous mode, the counter starts when the CCCR.INIT bit (CCCR <0>) is reset. A write to TOCV presets the counter to the value configured by the TOCC.TOP bits (TOCC <31:16>) 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 the TOCC.TOP bits (TOCC <31:16>). Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches zero, the Interrupt Flag, IR.TOO bit ( IR<18>) is set. In continuous mode, the counter is immediately restarted at the TOCC.TOP bits (TOCC <31:16>).

**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/re-synchronization 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.

### 33.5.4. Interrupts

The CAN has four interrupt sources:

- Two interrupt lines: LINE[1:0]

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.

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 CAN-FD IR register (EIP-00596) 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- AHB Bus Error Interrupt: BERR
- Debug Message Reception Interrupt: DBG

These two interrupt lines have dedicated bit in the INTFLAG register. The user must read the INTFLAG register to determine which interrupt condition is present.

The interrupt source can be 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). The interrupt request is generated when the interrupt flags is set, and the interrupt is enabled. The interrupt request remains active until the interrupt flags source of interrupt is cleared, or when the interrupt source is disabled. Refer to INTFLAG register for details on how to clear interrupt flags.

The interrupt flags are cleared in the case of a hardware reset condition detection.

### 33.5.5. Sleep Mode Operation

The CAN can be configured to operate in any sleep mode where AHB and GCLK clocks are provided.

The CAN has its own low power mode that may be used at any time. 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.

To leave low power mode, CSR bit in CCCR register must be written to 0. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT.

After reset, the GCLK_CAN and CLK_CAN_SYS clocks are not requested, except for each APB bus access. After the CAN initialization (CCCR.INIT =0), both GCLK_CAN and CLK_CAN_SYS clocks are requested as long as Clock Stop Request bit in the CC Control register is cleared (CCCR.CSR = 0), and stopped when Clock Stop Request bit in the CC Control register is set (CCCR.CSR = 1) and the Clock Stop Acknowledge bit is set (CCCR.CSA = 1).

In Standby sleep mode, when the RUNSTDBY bit is set (CTRLA.RUNSTDBY=1), the CAN IP is allowed to continue to run its operation.

When the RUNSTDBY bit is cleared (CTRLA.RUNSTDBY=0) and the Standby sleep mode is detected, a hardware mechanism will request the M_CAN peripheral to stop its operation, and enables the CAN low power mode automatically. The CCCR.CSR bit is automatically set one as long as the stop request operation is active

When all pending transmission requests have completed, the M_CAN waits until bus idle state is detected. Then the M_CAN sets then CCCR.INIT to one to prevent any further CAN transfers. The hardware stop request is acknowledged and CCCR.CSA bit is set one. The clocks requests are switched off.

When the system exits the Standby sleep mode, CSR bit in CCCR register must be written to 0. Afterwards, the application can restart CAN communication by resetting bit CCCR.INIT.

**Note:** In case of a heavily disturbed CAN bus, it may happen that idle state is never reached and CCCR.INIT is therefore not set by the M_CAN. This situation can be detected by polling PSR.ACT. In case there is a risk the M_CAN does not enter idle state, it is recommended to use the software procedure to put the M_CAN in low power mode, before the system goes to Standby sleep mode. In this way, the software can write CCCR.INIT = '1', which stops CAN communication of the M_CAN immediately, regardless whether there is a transmission/reception ongoing or not.

### 33.5.6. Synchronization

Due to the asynchronicity between the main clock domain (CLK_CAN_SYS) and the peripheral clock domain (GCLK_CAN) some registers are synchronized when written. In order to achieve a stable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

function of the M_CAN peripheral, the CLK_CAN_SYS clock must be faster or equal to the GCLK_CAN clock

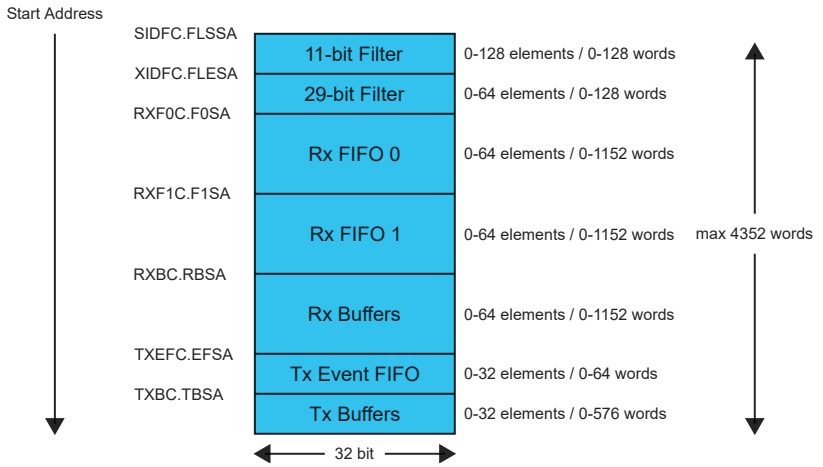
## 33.6. Message RAM

### 33.6.1. Message RAM Configuration

The Message RAM has a width of 32 bits. 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 through the RXESC.F0DS bits ( RXESC <2:0>), RXESC.F1DS bits (RXESC <6:4>), RXESC.RBDS bits ( RXESC <10:8>), and TXESC.TBDS bits (TXESC <2:0>).

**Figure 33-12.** 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).

**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.

### 33.6.2. 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 element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC.

### 33.6.3. 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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.

### 33.6.4. 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.

### 33.6.5. 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 bits (SIDFC <15:0>) plus the index of the filter element (0 ... 127).

### 33.6.6. 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 bits (XIDFC <15:0>) plus two times the index of the filter element (0...63).

### 33.6.7. REGISTER ACCESS PROTECTION

The following registers are write-protected:

- CTRLA - Control A Register
- CTRLB - Control B Register
- INTENCLR - Interrupt Enable Clear Register
- INTENSET - Interrupt Enable Set Register
- INTFLAGSET - Interrupt Flag Software Set Register

When the write protection is enabled, any attempt to write these registers will be discarded. These registers are marked as "write-protected".

M_CAN and TSU extension support specific write-protection mechanism. Configuration registers are accessible only when CCCR.INIT = 1 and CCCR.CCE are set. Some other registers are accessible when CCCR.CCE = 0. No interrupts or peripheral errors are generated when an access to a "protectedwrite" register is detected, and the protection is active.

### 33.6.8. Client Error Generation

A peripheral client error is generated when one of the following cases is detected:

- Any access to a non-implemented register, including locations from M_CAN and TSU register map.
- Write access to a write-protected or read-only register within 0x0000 - 0x0FFF address range
- Write access to a register with an ongoing synchronization (respective SYNCBUSY bit is set)

In addition to the peripheral slave error generation, an access (read or write) to a reserved register from M_CAN or TSU address map will set the IR.ARA interrupt flag, and if enabled, the interrupt is signaled via the assigned interrupt line (can_intreq_line[0] or can_intreq_line[1]). The interrupt flag is set only for accesses to reserved addresses where all four bytes are reserved.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 33.7. Peripheral Dependencies

**Table 33-8.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| CAN0 | 0x4402E000 | 35 : INT0<br>36 : INT1<br>37 : BERR/DBG | **AHB:**<br>CLKMSK[0]<br>MASK23 | PCHCTRL[6] | 17<br>STATUS0<br>PERID17 | | | PD_CORE_SW |
| CAN1 | 0x44030000 | 38 : INT0<br>39 : INT1<br>40 : BERR/DBG | **AHB:**<br>CLKMSK[0]<br>MASK24 | PCHCTRL[7] | 18<br>STATUS0<br>PERID18 | | | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 33.8. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | RUNSTDBY | | | | PRIV | | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CTRLB | 7:0 | | | | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | OFFSET[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 ... 0x27 | Reserved | | | | | | | | | |
| 0x28 | INTENCLR | 7:0 | | | | | | | BERR | DBG |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x2C | INTENSET | 7:0 | | | | | | | BERR | DBG |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x30 | INTFLAG | 7:0 | | | | | | | BERR | DBG |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | INTFLAGSET | 7:0 | | | | | | | BERR | DBG |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x38 ... 0x5B | Reserved | | | | | | | | | |
| 0x5C | SYNCBUSY | 7:0 | | | | | | | | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x60 ... 0xDF | Reserved | | | | | | | | | |
| 0xE0 | WPCTRL | 7:0 | | | | | | | WPLCK | WPEN |
| | | 15:8 | WPKEY[7:0] | | | | | | | |
| | | 23:16 | WPKEY[15:8] | | | | | | | |
| | | 31:24 | WPKEY[23:16] | | | | | | | |
| 0xE4 ... 0xFF | Reserved | | | | | | | | | |
| 0x0100 | CREL_MCAN | 7:0 | DAY[7:0] | | | | | | | |
| | | 15:8 | MON[7:0] | | | | | | | |
| | | 23:16 | SUBSTEP[3:0] | | | | YEAR[3:0] | | | |
| | | 31:24 | REL[3:0] | | | | STEP[3:0] | | | |
| 0x0104 | ENDN | 7:0 | ETV[7:0] | | | | | | | |
| | | 15:8 | ETV[15:8] | | | | | | | |
| | | 23:16 | ETV[23:16] | | | | | | | |
| | | 31:24 | ETV[31:24] | | | | | | | |
| 0x0108 | CUST | 7:0 | CUST[7:0] | | | | | | | |
| | | 15:8 | CUST[15:8] | | | | | | | |
| | | 23:16 | CUST[23:16] | | | | | | | |
| | | 31:24 | CUST[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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x010C | DBTP | 7:0 | DTSEG2[3:0] | | | | DSJW[3:0] | | | |
| | | 15:8 | | | | DTSEG1[4:0] | | | | |
| | | 23:16 | TDC | | | DBRP[4:0] | | | | |
| | | 31:24 | | | | | | | | |
| 0x0110 | TEST | 7:0 | RX | TX[1:0] | | LBCK | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0114 | RWD | 7:0 | WDC[7:0] | | | | | | | |
| | | 15:8 | WDV[7:0] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0118 | CCCR | 7:0 | TEST | DAR | MON | CSR | CSA | ASM | CCE | INIT |
| | | 15:8 | | TXP | EFBI | PXHD | WMM | UTSU | BRSE | FDOE |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x011C | NBTP | 7:0 | | NTSEG2[6:0] | | | | | | |
| | | 15:8 | NTSEG1[7:0] | | | | | | | |
| | | 23:16 | NBRP[7:0] | | | | | | | |
| | | 31:24 | NSJW[6:0] | | | | | | | NBRP[8] |
| 0x0120 | TSCC | 7:0 | | | | | | | TSS[1:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | TCP[3:0] | | | |
| | | 31:24 | | | | | | | | |
| 0x0124 | TSCV | 7:0 | TSC[7:0] | | | | | | | |
| | | 15:8 | TSC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0128 | TOCC | 7:0 | | | | | | TOS[1:0] | | ETOC |
| | | 15:8 | | | | | | | | |
| | | 23:16 | TOP[7:0] | | | | | | | |
| | | 31:24 | TOP[15:8] | | | | | | | |
| 0x012C | TOCV | 7:0 | TOC[7:0] | | | | | | | |
| | | 15:8 | TOC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0130 ... 0x013F | Reserved | | | | | | | | | |
| 0x0140 | ECR | 7:0 | TEC[7:0] | | | | | | | |
| | | 15:8 | RP | REC[6:0] | | | | | | |
| | | 23:16 | CEL[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0144 | 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 | | | | | | | | |
| 0x0148 | TDCR | 7:0 | | TDCF[6:0] | | | | | | |
| | | 15:8 | | TDCO[6:0] | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x014C ... 0x014F | Reserved | | | | | | | | | |
| 0x0150 | 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 |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0154 | 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 |
| 0x0158 | 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 |
| 0x015C | ILE | 7:0 | | | | | | | EINT1 | EINT0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0160 ... 0x017F | Reserved | | | | | | | | | |
| 0x0180 | GFC | 7:0 | | | ANFS[1:0] | | ANFE[1:0] | | RRFS | RRFE |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0184 | SIDFC | 7:0 | FLSSA[5:0] | | | | | | | |
| | | 15:8 | FLSSA[13:6] | | | | | | | |
| | | 23:16 | LSS[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0188 | XIDFC | 7:0 | FLESA[5:0] | | | | | | | |
| | | 15:8 | FLESA[13:6] | | | | | | | |
| | | 23:16 | | LSE[6:0] | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x018C ... 0x018F | Reserved | | | | | | | | | |
| 0x0190 | XIDAM | 7:0 | EIDM[7:0] | | | | | | | |
| | | 15:8 | EIDM[15:8] | | | | | | | |
| | | 23:16 | EIDM[23:16] | | | | | | | |
| | | 31:24 | | | | EIDM[28:24] | | | | |
| 0x0194 | HPMS | 7:0 | MSI[1:0] | | BIDX[5:0] | | | | | |
| | | 15:8 | FLST | FIDX[6:0] | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0198 | 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 |
| 0x019C | 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 |
| 0x01A0 | RXF0C | 7:0 | F0SA[7:0] | | | | | | | |
| | | 15:8 | F0SA[15:8] | | | | | | | |
| | | 23:16 | | F0S[6:0] | | | | | | |
| | | 31:24 | F0OM | F0WM[6:0] | | | | | | |
| 0x01A4 | RXF0S | 7:0 | | F0FL[6:0] | | | | | | |
| | | 15:8 | | | F0GI[5:0] | | | | | |
| | | 23:16 | | | F0PI[5:0] | | | | | |
| | | 31:24 | | | | | | | RF0L | F0F |
| 0x01A8 | RXF0A | 7:0 | | | F0AI[5:0] | | | | | |
| | | 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01AC | RXBC | 7:0 | RBSA[7:0] | | | | | | | |
| | | 15:8 | RBSA[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01B0 | RXF1C | 7:0 | F1SA[5:0] | | | | | | | |
| | | 15:8 | F1SA[13:6] | | | | | | | |
| | | 23:16 | | F1S[6:0] | | | | | | |
| | | 31:24 | F1OM | F1WM[6:0] | | | | | | |
| 0x01B4 | RXF1S | 7:0 | | F1FL[6:0] | | | | | | |
| | | 15:8 | | | F1GI[5:0] | | | | | |
| | | 23:16 | | | F1PI[5:0] | | | | | |
| | | 31:24 | DMS[1:0] | | | | | | RF1L | F1F |
| 0x01B8 | RXF1A | 7:0 | | | F1AI[5:0] | | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01BC | RXESC | 7:0 | | F1DS[2:0] | | | | F0DS[2:0] | | |
| | | 15:8 | | | | | | RBDS[2:0] | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01C0 | TXBC | 7:0 | TBSA[7:0] | | | | | | | |
| | | 15:8 | TBSA[15:8] | | | | | | | |
| | | 23:16 | | | NDTB[5:0] | | | | | |
| | | 31:24 | | TFQM | TFQS[5:0] | | | | | |
| 0x01C4 | TXFQS | 7:0 | | | TFFL[5:0] | | | | | |
| | | 15:8 | | | | TFGI[4:0] | | | | |
| | | 23:16 | | | TFQF | TFQPI[4:0] | | | | |
| | | 31:24 | | | | | | | | |
| 0x01C8 | TXESC | 7:0 | | | | | | | TBDS[2:0] | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01CC | 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 |
| 0x01D0 | 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 |
| 0x01D4 | 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 |
| 0x01D8 | 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 |
| 0x01DC | 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 |
| 0x01E0 | 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 |
| 0x01E4 | 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 |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x01E8 ... 0x01EF | Reserved | | | | | | | | | |
| 0x01F0 | TXEFC | 7:0 | EFSA[5:0] | | | | | | | |
| | | 15:8 | EFSA[13:6] | | | | | | | |
| | | 23:16 | | | EFS[5:0] | | | | | |
| | | 31:24 | | | EFWM[5:0] | | | | | |
| 0x01F4 | TXEFS | 7:0 | | | EFFL[5:0] | | | | | |
| | | 15:8 | | | | EFGI[4:0] | | | | |
| | | 23:16 | | | | EFPI[4:0] | | | | |
| | | 31:24 | | | | | | | TEFL | EFF |
| 0x01F8 | TXEFA | 7:0 | | | | EFAI[4:0] | | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x01FC ... 0x025F | Reserved | | | | | | | | | |
| 0x0260 | CREL_TSU | 7:0 | DAY[7:0] | | | | | | | |
| | | 15:8 | MON[7:0] | | | | | | | |
| | | 23:16 | SUBSTEP[3:0] | | | | YEAR[3:0] | | | |
| | | 31:24 | REL[3:0] | | | | STEP[3:0] | | | |
| 0x0264 | TSCFG | 7:0 | | | | | | SCP | TBCS | TSUE |
| | | 15:8 | TBPRE[7:0] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0268 | TSS1 | 7:0 | TSL[7:0] | | | | | | | |
| | | 15:8 | TSL[15:8] | | | | | | | |
| | | 23:16 | TSN[7:0] | | | | | | | |
| | | 31:24 | TSN[15:8] | | | | | | | |
| 0x026C | TSS2 | 7:0 | | | | | TSP[3:0] | | | |
| | | 15:8 | ITBG[1:0] | | NTSG[1:0] | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0270 | TS0 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x0274 | TS1 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x0278 | TS2 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x027C | TS3 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x0280 | TS4 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x0284 | TS5 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0288 | TS6 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x028C | TS7 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x0290 | TS8 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x0294 | TS9 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x0298 | TS10 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x029C | TS11 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x02A0 | TS12 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x02A4 | TS13 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x02A8 | TS14 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x02AC | TS15 | 7:0 | TS[7:0] | | | | | | | |
| | | 15:8 | TS[15:8] | | | | | | | |
| | | 23:16 | TS[23:16] | | | | | | | |
| | | 31:24 | TS[31:24] | | | | | | | |
| 0x02B0 | ATB | 7:0 | TB[7:0] | | | | | | | |
| | | 15:8 | TB[15:8] | | | | | | | |
| | | 23:16 | TB[23:16] | | | | | | | |
| | | 31:24 | TB[31:24] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.1. Control A Register

**Name:** CTRLA
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** RW

**Table 33-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | RUNSTDBY | | | | PRIV | | SWRST |
| Access | | R/W | | | | R/W | | R/S/HC |
| Reset | | 0 | | | | 0 | | 0 |

**Bit 6 – RUNSTDBY** Run during standby

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | The module is internally disabled and stops the operation when all bus transactions are completed. |
| 1 | ENABLE | The module continues the normal operation in Standby sleep mode. |

**Bit 2 – PRIV** Priveged Access

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | The CAN registers are accessible in privilege and non-privilege mode. |
| 1 | ENABLE | The CAN registers are accessible in privilege mode |

**Bit 0 – SWRST** Software Reset

Writing a '1' to this bit will reset all APB registers, including the M_CAN and TSU registers.
The bit stays high until reset completes. Setting this bit also sets the SYNCBUSY.SWRST to 1.
SYNCBUSY.SWRST stays 1 until reset sequence completes.

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No Action. |
| 1 | RESET | Reset all APB registers. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.2. Control B Register

**Name:** CTRLB
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** R/W

**Table 33-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | OFFSET[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 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bits 23:16 – OFFSET[7:0]** Message RAM Base Address Offset.
This bitfield value represents the 8 bits offset of the memory base address (bits [23:16]). The base address is calculated using the following formula:
Base_Address = 0x20 << 24 + OFFSET << 16.
**Note:** It is recommended to write this register only when the Change Configuration bit in the CC Control register is set (CCCR.CCE = 1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.3. Interrupt Enable Clear Register

**Name:** INTENCLR
**Offset:** 0x028
**Reset:** 0x00000000
**Property:** R/K

**Table 33-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BERR | DBG |
| Access | | | | | | | R/K | R/K |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – BERR** Bus Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Bus Error Interrupt Enable bit, which disables the Bus Error interrupt.

| Value | Description |
|---|---|
| 0 | The Bus Error Interrupt is disabled. |
| 1 | The Bus Error Interrupt is enabled, and an interrupt request is generated when the Bus Error Message Interrupt Flag is set. |

**Bit 0 – DBG** Debug Message Interrupt Disable
Writing '0' to this bit has no effect.
Writing a '1' to this bit will clear the Debug Message Interrupt Enable bit, which disables the Debug Message Interrupt.

| Value | Description |
|---|---|
| 0 | The Debug Message Interrupt is disabled. |
| 1 | The Debug Message Interrupt is enabled, and an interrupt request is generated when the Debug Message Interrupt Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.4. Interrupt Enable Set Register

**Name:** INTENSET
**Offset:** 0x02C
**Reset:** 0x00000000
**Property:** R/S

**Table 33-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BERR | DBG |
| Access | | | | | | | R/S | R/S |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – BERR** Bus Error Interrupt Enable
Writing '0' to this bit has no effect.
Writing '1' to this bit will clear the Bus Error Interrupt Enable bit, which disables the Bus Error interrupt.

| Value | Description |
|---|---|
| 0 | The Bus Error Interrupt is disabled. |
| 1 | The Bus Error Interrupt is enabled, and an interrupt request is generated when the Bus Error Message Interrupt Flag is set. |

**Bit 0 – DBG** Debug Message Interrupt Enable
Writing '0' to this bit has no effect.
Writing a '1' to this bit will clear the Debug Message Interrupt Enable bit, which disables the Debug Message Interrupt.

| Value | Description |
|---|---|
| 0 | The Debug Message Interrupt is disabled. |
| 1 | The Debug Message Interrupt is enabled, and an interrupt request is generated when the Debug Message Interrupt Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.5. Interrupt Flag Status and Clear Register

**Name:** INTFLAG
**Offset:** 0x030
**Reset:** 0x00000000
**Property:** R/K

**Table 33-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BERR | DBG |
| Access | | | | | | | R/K/HS | R/K/HS |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – BERR** AHB Bus Error Detection
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the AHB Bus Error Detection Interrupt Flag.

| Value | Description |
|---|---|
| 0 | The AHB Bus Error Detection Flag is cleared. |
| 1 | The AHB Bus Error Detection Flag is set. |

**Bit 0 – DBG** Debug Message Reception
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Debug Message Interrupt Flag.

| Value | Description |
|---|---|
| 0 | The Debug Message Flag is cleared. |
| 1 | The Debug Message Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.6. Interrupt Flag Software Set Register

**Name:** INTFLAGSET
**Offset:** 0x034
**Reset:** 0x00000000
**Property:** R/S

**Table 33-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | BERR | DBG |
| Access | | | | | | | R/S | R/S |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – BERR** AHB Bus Error Detection Interrupt Flag Set
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the AHB Bus Error Detection Interrupt Flag.

| Value | Description |
|---|---|
| 0 | The AHB Bus Error Detection Flag is cleared. |
| 1 | The AHB Bus Error Detection Flag is set. |

**Bit 0 – DBG** Debug Message Interrupt Flag Set
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Debug Message Interrupt Flag.

| Value | Description |
|---|---|
| 0 | The Debug Message Flag is cleared. |
| 1 | The Debug Message Flag is set. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.7. SYNCBUSY Register

**Name:** SYNCBUSY
**Offset:** 0x05C
**Reset:** 0x00000000
**Property:** R

**Table 33-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | | | | | | R/HS/HC |
| Reset | | | | | | | | 0 |

**Bit 0 – SWRST** Software Reset Synchronization Busy

| Value | Description |
|---|---|
| 0 | SWRST synchronization is not busy. |
| 1 | SWRST synchronization is busy. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.8. Write Protection Control Register

**Name:** WPCTRL
**Offset:** 0x0E0
**Reset:** 0x00000000
**Property:** RW

**Table 33-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | WPLCK | WPEN |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 31:8 – WPKEY[23:0]** Write Protection Key
Writing a value other than WPCTRL_KEY to this field cancels the write operation this register and generates a slave bus error. This field always returns 0 on read.

**Bit 1 – WPLCK** Write Lock Bit

**Bit 0 – WPEN** Write Protection Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.9. Core Release

**Name:** CREL_MCAN
**Offset:** 0x100
**Reset:** 0x33000000
**Property:** R

**Table 33-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | 1 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SUBSTEP[3:0] | | | | YEAR[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 |
|---|---|---|---|---|---|---|---|---|
| | MON[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 |
|---|---|---|---|---|---|---|---|---|
| | DAY[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:28 – REL[3:0]** Core Release

**Bits 27:24 – STEP[3:0]** Step of Core Release

**Bits 23:20 – SUBSTEP[3:0]** Sub-step of Core Release

**Bits 19:16 – YEAR[3:0]** Timestamp Year

**Bits 15:8 – MON[7:0]** Timestamp Month

**Bits 7:0 – DAY[7:0]** Timestamp Day


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.10. Endian

**Name:** ENDN
**Offset:** 0x104
**Reset:** 0x87654321
**Property:** R

**Table 33-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.11. Customer Register

**Name:** CUST
**Offset:** 0x108
**Reset:** 0x00000000
**Property:** RW

**Table 33-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CUST[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 |
|---|---|---|---|---|---|---|---|---|
| | CUST[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 |
|---|---|---|---|---|---|---|---|---|
| | CUST[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 |
|---|---|---|---|---|---|---|---|---|
| | CUST[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CUST[31:0]** Customer Register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.12. Fast Bit Timing and Prescaler

**Name:** DBTP
**Offset:** 0x10C
**Reset:** 0x00000A33
**Property:** RW

This register is write-restricted and only writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

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_CANx periods. time quantum ($t_q$) = (DBRP + 1) mtq. Therefore the length of the bit time is [DTSEG1 + DTSEG2 + 3] tq where DTSEG1 and DTSEG2 are programmed values in the DBTP register.

**Note:**
With a GCLK_CANx 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 the DBTP register must be higher or equal to the bit rate configured for the arbitration phase via the NBTP register.

**Table 33-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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** Tranceiver Delay Compensation

| Value | Description |
|---|---|
| 0 | Transceiver Delay Compensation disabled. |
| 1 | Transceiver Delay Compensation enabled. |

**Bits 20:16 – DBRP[4:0]** Data Baud Rate Prescaler


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0x00 – 0x1F | The value by which the GCLK_CANx 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]** Data 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 PROPAGATION TIME SEGMENT (PROP_SEG) and PHASE BUFFER SEGMENT1 (PHASE_SEG1). |

**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 BUFFER SEGMENT2 (PHASE_SEG2). |

**Bits 3:0 – DSJW[3:0]** Data (Re)Synchronization 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.

### 33.8.13. Test

**Name:** TEST
**Offset:** 0x110
**Reset:** 0x00000000
**Property:** RW

**Table 33-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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
This bit reflects the actual value of pin CANx_RX. The read value can be interpret as follows.

| Value | Description |
|---|---|
| 0 | The CAN bus is dominant (CANx_RX = 0). |
| 1 | The CAN bus is recessive (CANx_RX = 1). |

**Bits 6:5 – TX[1:0]** Control of Transmit Pin
This field defines the control of the transmit pin.

| Value | Name | Description |
|---|---|---|
| 0 | CORE | Reset value, CAN_TX controlled by CAN core, updated at the end of CAN bit time. |
| 1 | SAMPLE | Sample Point can be monitored at pin CAN_TX. |
| 2 | DOMINANT | Dominant (0) level at pin CAN_TX |
| 3 | 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.

### 33.8.14. RAM Watchdog

**Name:** RWD
**Offset:** 0x114
**Reset:** 0x00000000
**Property:** RW

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

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 WDC bits (RWD <7:0>). The counter is reloaded with WDC bits (RWD <7:0>) 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 bit (IR<26>) is set.

**Table 33-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.15. CC Control

**Name:** CCCR
**Offset:** 0x118
**Reset:** 0x00000001
**Property:** RW

**Table 33-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | WMM | UTSU | BRSE | FDOE |
| Access | | RW | RW | RW | R/W | R/W | RW | RW |
| Reset | | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TEST | DAR | MON | CSR | CSA | ASM | CCE | INIT |
| Access | RW | RW | RW | RW | R | RW | RW | 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 both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

| 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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 11 – WMM** Wide Message Marker

**Bit 10 – UTSU** Use Timestamping Unit

| Value | Description |
|---|---|
| 0 | Use the internal Timestamp counter |
| 1 | Use the external Timestamp Unit |

**Bit 9 – BRSE** Bit Rate Switch Enable

**Note:** When CAN FD operation is disabled (i.e.,CCCR.FDOE bit ( CCCR <8>) = 0), BRSE is not evaluated.

| Value | Description |
|---|---|
| 0 | Bit rate switching for transmissions disabled. |
| 1 | Bit rate switching for transmissions enabled. |

**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 allowed only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

| 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 writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

| 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 allowed only both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

| 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 CCCR.INIT bit (CCCR <0>) and then the CCCR.CSA bit (CCCR <3>) will be set after all pending transfer requests have been completed and the CAN bus reached idle. |

**Bit 3 – CSA** Clock Stop Acknowledge


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | No clock stop acknowledged. |
| 1 | CAN may be set in power down by stopping CLK_CANx_AHB and GCLK_CANx. |

**Bit 2 – ASM** 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 allowed only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

| 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 CCCR.INIT bit (CCCR <0>) is set.

| Value | Description |
|---|---|
| 0 | The CPU has no write access to the protected configuration registers. |
| 1 | The CPU has write access to the protected configuration registers (while CCCR.INIT bit (CCCR <0>) =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 the INIT bit can be read back. The programmer has to assure that the previous value written to the INIT bit has been accepted by reading the INIT bit before setting the INIT bit to a new value.

| Value | Description |
|---|---|
| 0 | Normal Operation. |
| 1 | Initialization is started. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.16. Nominal Bit Timing and Prescaler

**Name:** NBTP
**Offset:** 0x11C
**Reset:** 0x06000A03
**Property:** RW

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

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_CANx periods. tq = (NBRP + 1) mtq.

Therefore the length of the bit time is [NTSEG1 + NTSEG2 + 3] tq, where NTSEG1 and NTSEG2 are programmed values in the NBTP register.

**Note:** With a CAN clock (GCLK_CANx) of 8MHz, the reset value 0x06000A03 configures the CAN for a bit rate of 500 kBits/s.

**Table 33-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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)Synchronization 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 GCLK_CANx 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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 15:8 – NTSEG1[7:0]** Nominal Time Segment Before Sample Point

| 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 PROPAGATION TIME SEGMENT (PROP_SEG) and PHASE BUFFER SEGMENT1 (PHASE_SEG1). |

**Bits 6:0 – NTSEG2[6:0]** Nominal 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 BUFFER SEGMENT2 (PHASE_SEG2). |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.17. Timestamp Counter Configuration

**Name:** TSCC
**Offset:** 0x120
**Reset:** 0x00000000
**Property:** RW

This register is write-restricted and writable only if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

**Table 33-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | TCP[3:0] | | | |
| Access | | | | | RW | RW | RW | RW |
| 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 | | | | | | | RW | RW |
| 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 |
|---|---|---|
| 0 | ZERO | Timestamp counter value always 0x0000 |
| 1 | INC | Timestamp counter value incremented by TCP |
| 2 | EXT | External timestamp counter value used |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.18. Timestamp Counter Value

**Name:** TSCV
**Offset:** 0x124
**Reset:** 0x00000000
**Property:** R

**Notes:**

1. A write access to TSCV clears the Timestamp Counter value.
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.

**Table 33-26.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 bitfield TSCC.TSS (TSCC<1:0>) = 0x1, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of bitfield TSCC.TCP (TSCC<19:16>) . A wrap around sets interrupt flag IR.TSW (IR<16>) .


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.19. Timeout Counter Configuration

**Name:** TOCC
**Offset:** 0x128
**Reset:** 0xFFFF0000
**Property:** RW

This register is write-restricted and writable if both CCCR.CCE bit (CCCR <1>) and CCCR.INIT bit (CCCR <0>) are set.

**Table 33-27.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 register 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 |
|---|---|---|
| 0 | CONT | Continuout operation |
| 1 | TXEF | Timeout controlled by TX Event FIFO |
| 2 | RXF0 | Timeout controlled by Rx FIFO 0 |
| 3 | RXF1 | Timeout controlled by Rx FIFO 1 |

**Bit 0 – ETOC** Enable Timeout Counter

| Value | Description |
|---|---|
| 0 | Timeout Counter disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Timeout Counter enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.20. Timeout Counter Value

**Name:** TOCV
**Offset:** 0x12C
**Reset:** 0x0000FFFF
**Property:** RW

**Note:** A write access to TOCV reloads the Timeout Counter with the value of bit TOCC.TOP (TOCC<31:16>).

**Table 33-28.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.21. Error Counter

**Name:** ECR
**Offset:** 0x140
**Reset:** 0x00000000
**Property:** R

**Note:** When CCCR.ASM bit (CCCR<2>) is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

**Table 33-29.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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; and 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.

### 33.8.22. Protocol Status

**Name:** PSR
**Offset:** 0x144
**Reset:** 0x00000707
**Property:** R

**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 PSR.DLEC 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 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 wait time after the resetting of CCCR.INIT, each time a sequence of 11 recessive bits is monitored, a Bit0 Error code is written to PSR.LEC. This enables the CPU to readily check the status of CAN bus (whether bus is stuck at dominant level or continuously disturbed). ECR.REC is used to count these sequences.

**Table 33-30.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 CANx_TX to CANx_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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 14 – PXE** Protocol Exception Event

This field is cleared on read access. A recessive “reserved bit” following a recessive FDF bit is the example of Protocol Exception.

| Value | Description |
|---|---|
| 0 | No protocol exception event occurred since last read access. |
| 1 | Protocol exception event occurred. |

**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 Phase 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.

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No Error |
| 1 | STUFF | Stuff Error |
| 2 | FORM | Form Error |
| 3 | ACK | Ack Error |
| 4 | BIT1 | Bit1 Error |
| 5 | BIT0 | Bit0 Error |
| 6 | CRC | CRC Error |
| 7 | NC | No Change |

**Bit 7 – BO** Bus_Off Status

| Value | Description |
|---|---|
| 0 | The CAN is not 'bus off' state. |
| 1 | The CAN is in 'bus off' state. |

**Bit 6 – EW** 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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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 |
|---|---|---|
| 0 | SYNC | Node is synchronizing on CAN communication |
| 1 | IDLE | Node is neither receiver nor transmitter |
| 2 | RX | Node is operating as receiver |
| 3 | TX | Node is operating as transmitter |

**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.

### 33.8.23. Extended ID Filter Configuration

**Name:** TDCR
**Offset:** 0x148
**Reset:** 0x00000000
**Property:** R/W

**Table 33-31.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 CANx_TX to CANx_RX and the secondary sample point. Valid values are 0 to 127 mtq. |

**Bits 6:0 – TDCF[6:0]** Transmitter Delay Compensation Filter Length

| Value | Description |
|---|---|
| 0x00 – 0x7F | Defines the minimum value for the SSP position, dominant edges on CANx_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.

### 33.8.24. Interrupt

**Name:** IR
**Offset:** 0x150
**Reset:** 0x00000000
**Property:** RW

**Note:** Interrupt flags must be cleared and then read back to confirm the clear before exiting the ISR to avoid double interrupts.

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.

**Table 33-32.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 detected. |
| 1 | Protocol error in data phase detected (i.e., PSR.DLEC bits ( PSR<10:8>) != 0 and PSR.DLEC bits ( PSR<10:8>) != 0 ). |

**Bit 27 – PEA** Protocol Error in Arbitration Phase


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | No protocol error in arbitration phase detected. |
| 1 | Protocol error in arbitration phase detected (i.e., PSR.LEC bits ( PSR<2:0>) != 0 and PSR.LEC bits ( PSR<2:0>) != 0 ). |

**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

| Value | Description |
|---|---|
| 0 | 'bus off' status unchanged. |
| 1 | 'bus off' status changed. |

**Bit 24 – EW** 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

**Bit 20 – BEC** Bit Error Corrected

**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. 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 must clear 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 not-empty. |
| 1 | Tx FIFO empty. |

**Bit 10 – TCF** Transmission Cancellation Finished

| Value | Description |
|---|---|
| 0 | Transmission cancellation not 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 FIFO 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.

### 33.8.25. Interrupt Enable

**Name:** IE
**Offset:** 0x154
**Reset:** 0x00000000
**Property:** RW

**Table 33-33.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 28 – PEDE** Protocol Error in Data Phase Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 27 – PEAE** Protocol Error in Arbitration Phase Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 26 – WDIE** Watchdog Interrupt Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Interrupt enabled. |

**Bit 25 – BOE** Bus_Off Status Interrupt Enable

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 24 – EWE** 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 15 – TEFLE** Tx Event FIFO Element Lost Interrupt Enable

| 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** Timestamp 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 FIFO Full Interrupt Enable


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Interrupt disabled. |
| 1 | Interrupt enabled. |

**Bit 5 – RF1WE** Rx FIFO 1 Watermark Reached Interrupt Enable

| 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.

### 33.8.26. Interrupt Line Select

**Name:** ILS
**Offset:** 0x158
**Reset:** 0x00000000
**Property:** RW

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from IR to one of the two module interrupt lines.

**Table 33-34.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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 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 Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 26 – WDIL** Watchdog Interrupt 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 25 – BOL** Bus_Off Status Interrupt Line

| Value | Description |
|---|---|
| 0 | Interrupt assigned to CAN interrupt line 0. |
| 1 | Interrupt assigned to CAN interrupt line 1. |

**Bit 24 – EWL** 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


© 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 15 – TEFLL** Tx Event FIFO Element Lost Interrupt Line

| 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** Timestamp 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 FIFO Full 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 5 – RF1WL** Rx FIFO 1 Watermark Reached Interrupt Line

| 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.

### 33.8.27. Interrupt Line Enable

**Name:** ILE
**Offset:** 0x15C
**Reset:** 0x00000000
**Property:** R/W

**Table 33-35.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | EINT1 | EINT0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bit 1 – EINT1** Enable Interrupt Line 1

| Value | Description |
|---|---|
| 0 | CAN interrupt line 1 disabled. |
| 1 | CAN interrupt line 1 enabled. |

**Bit 0 – EINT0** Enable Interrupt Line 0

| Value | Description |
|---|---|
| 0 | CAN interrupt line 0 disabled. |
| 1 | CAN interrupt line 0 enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.28. Global Filter Configuration

**Name:** GFC
**Offset:** 0x180
**Reset:** 0x00000000
**Property:** RW

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

**Table 33-36.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|
| 0 | RXF0 | Accept in Rx FIFO 0 |
| 1 | RXF1 | Accept in Rx FIFO 1 |
| 2 | 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 |
|---|---|---|
| 0 | RXF0 | Accept in Rx FIFO 0 |
| 1 | RXF1 | Accept in Rx FIFO 1 |
| 2 | REJECT | Reject |

**Bit 1 – RRFS** Reject Remote Frames Standard


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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

| 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.

### 33.8.29. Standard ID Filter Configuration

**Name:** SIDFC
**Offset:** 0x184
**Reset:** 0x00000000
**Property:** R/W

This register is write-restricted and only writable if bit fields CCCR.CCE= 1 and CCCR.INIT= 1.

**Table 33-37.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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[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 |
|---|---|---|---|---|---|---|---|---|
| | FLSSA[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 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 15:2 – FLSSA[13:0]** 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.

### 33.8.30. Extended ID Filter Configuration

**Name:** XIDFC
**Offset:** 0x188
**Reset:** 0x00000000
**Property:** RW

**Table 33-38.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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[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 |
|---|---|---|---|---|---|---|---|---|
| | FLESA[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 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 15:2 – FLESA[13:0]** 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.

### 33.8.31. Extended ID AND Mask

**Name:** XIDAM
**Offset:** 0x190
**Reset:** 0x1FFFFFFF
**Property:** R/W

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

**Table 33-39.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.32. High Priority Message Status

**Name:** HPMS
**Offset:** 0x194
**Reset:** 0x00000000
**Property:** R

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.

**Table 33-40.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|
| 0 | NONE | No FIFO selected |
| 1 | LOST | FIFO message lost |
| 2 | FIFO0 | Message stored in FIFO 0 |
| 3 | FIFO1 | Message stored in FIFO 1 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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.

### 33.8.33. New Data 1

**Name:** NDAT1
**Offset:** 0x198
**Reset:** 0x00000000
**Property:** RW

**Table 33-41.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.34. New Data 2

**Name:** NDAT2
**Offset:** 0x19C
**Reset:** 0x00000000
**Property:** RW

**Table 33-42.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.35. Rx FIFO 0 Configuration

**Name:** RXF0C
**Offset:** 0x1A0
**Reset:** 0x00000000
**Property:** R/W

**Table 33-43.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 bit ( IR <1>)). |
| >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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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.

### 33.8.36. Rx FIFO 0 Status

**Name:** RXF0S
**Offset:** 0x1A4
**Reset:** 0x00000000
**Property:** R

**Table 33-44.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 bit = '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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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.

### 33.8.37. Rx FIFO 0 Acknowledge

**Name:** RXF0A
**Offset:** 0x1A8
**Reset:** 0x00000000
**Property:** RW

**Table 33-45.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.38. Rx Buffer Configuration

**Name:** RXBC
**Offset:** 0x1AC
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

**Table 33-46.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.39. Rx FIFO 1 Configuration

**Name:** RXF1C
**Offset:** 0x1B0
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 1.

**Table 33-47.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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[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 |
|---|---|---|---|---|---|---|---|---|
| | F1SA[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 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 bit (IR <5>)). |
| >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 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 – 64 | Number of Rx FIFO 1 elements. |
| >64 | Values greater than 64 are interpreted as 64. |

**Bits 15:2 – F1SA[13:0]** 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.

### 33.8.40. Rx FIFO 1 Status

**Name:** RXF1S
**Offset:** 0x1B4
**Reset:** 0x00000000
**Property:** Read-only

**Table 33-48.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|
| 0 | IDLE | Idle state |
| 1 | DBGA | Debug message A received |
| 2 | DBGB | Debug message A/B received |
| 3 | DBGC | Debug message A/B/C received, DMA request set |

**Bit 25 – RF1L** Rx FIFO 1 Message Lost
This bit is a copy of interrupt flag IR.RF1L bit (IR <7>). When IR.RF1L bit (IR <7>) is reset, this bit is also reset.
Overwriting the oldest message when RXF1C.F1OM bit (RXF1C <31>) = '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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 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.

**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.

### 33.8.41. Rx FIFO 1 Acknowledge

**Name:** RXF1A
**Offset:** 0x1B8
**Reset:** 0x00000000
**Property:** RW

**Table 33-49.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.42. Rx Buffer / FIFO Element Size Configuration

**Name:** RXESC
**Offset:** 0x1BC
**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.

**Table 33-50.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|
| 0 | DATA8 | 8 byte data field |
| 1 | DATA12 | 12 byte data field |
| 2 | DATA16 | 16 byte data field |
| 3 | DATA20 | 20 byte data field |
| 4 | DATA24 | 24 byte data field |
| 5 | DATA32 | 32 byte data field |
| 6 | DATA48 | 48 byte data field |
| 7 | DATA64 | 64 byte data field |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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 |
|---|---|---|
| 0 | DATA8 | 8 byte data field |
| 1 | DATA12 | 12 byte data field |
| 2 | DATA16 | 16 byte data field |
| 3 | DATA20 | 20 byte data field |
| 4 | DATA24 | 24 byte data field |
| 5 | DATA32 | 32 byte data field |
| 6 | DATA48 | 48 byte data field |
| 7 | 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 |
|---|---|---|
| 0 | DATA8 | 8 byte data field |
| 1 | DATA12 | 12 byte data field |
| 2 | DATA16 | 16 byte data field |
| 3 | DATA20 | 20 byte data field |
| 4 | DATA24 | 24 byte data field |
| 5 | DATA32 | 32 byte data field |
| 6 | DATA48 | 48 byte data field |
| 7 | DATA64 | 64 byte data field |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.43. Tx Buffer Configuration

**Name:** TXBC
**Offset:** 0x1C0
**Reset:** 0x00000000
**Property:** Write-restricted

**Table 33-51.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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 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.

### 33.8.44. Tx FIFO / Queue Status

**Name:** TXFQS
**Offset:** 0x1C4
**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. For 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.

**Table 33-52.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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.

### 33.8.45. Tx Buffer Element Size Configuration

**Name:** TXESC
**Offset:** 0x1C8
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE bit (CCCR <1>) = 1 and CCCR.INIT bit (CCCR <0>) = 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.

**Table 33-53.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | TBDS[2:0] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**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 |
|---|---|---|
| 0 | DATA8 | 8 byte data field |
| 1 | DATA12 | 12 byte data field |
| 2 | DATA16 | 16 byte data field |
| 3 | DATA20 | 20 byte data field |
| 4 | DATA24 | 24 byte data field |
| 5 | DATA32 | 32 byte data field |
| 6 | DATA48 | 48 byte data field |
| 7 | DATA64 | 64 byte data field |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.46. Tx Buffer Request Pending

**Name:** TXBRP
**Offset:** 0x1CC
**Reset:** 0x00000000
**Property:** R

**Note:** The 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.

**Table 33-54.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- 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. |
| 1 | Transmission request pending. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.47. Tx Buffer Add Request

**Name:** TXBAR
**Offset:** 0x1D0
**Reset:** 0x00000000
**Property:** -

**Note:** If an add request is applied for a Tx Buffer with pending transmission request (the corresponding TXBRP bit is already set), this add request is ignored.

**Table 33-55.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.48. Tx Buffer Cancellation Request

**Name:** TXBCR
**Offset:** 0x1D4
**Reset:** 0x00000000
**Property:** -

**Table 33-56.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.49. Tx Buffer Transmission Occurred

**Name:** TXBTO
**Offset:** 0x1D8
**Reset:** 0x00000000
**Property:** Read-only

**Table 33-57.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.50. Tx Buffer Cancellation Finished

**Name:** TXBCF
**Offset:** 0x1DC
**Reset:** 0x00000000
**Property:** Read-only

**Table 33-58.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.51. Tx Buffer Transmission Interrupt Enable

**Name:** TXBTIE
**Offset:** 0x1E0
**Reset:** 0x00000000
**Property:** -

**Table 33-59.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.52. Tx Buffer Cancellation Finished Interrupt Enable

**Name:** TXBCIE
**Offset:** 0x1E4
**Reset:** 0x00000000
**Property:** -

**Table 33-60.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.53. Tx Event FIFO Configuration

**Name:** TXEFC
**Offset:** 0x1F0
**Reset:** 0x00000000
**Property:** Write-restricted

This register is write-restricted and only writable if bit fields CCCR.CCE = 1 and CCCR.INIT = 1.

**Table 33-61.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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[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 |
|---|---|---|---|---|---|---|---|---|
| | EFSA[5:0] | | | | | | | |
| Access | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 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 bit (IR <13>)). |
| >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 15:2 – EFSA[13:0]** 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.

### 33.8.54. Tx Event FIFO Status

**Name:** TXEFS
**Offset:** 0x1F4
**Reset:** 0x00000000
**Property:** Read-only

**Table 33-62.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 33.8.55. Tx Event FIFO Acknowledge

**Name:** TXEFA
**Offset:** 0x1F8
**Reset:** 0x00000000
**Property:** -

**Table 33-63.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.56. TSU CREL

**Name:** CREL_TSU
**Offset:** 0x260
**Reset:** 0x10000000
**Property:** R

**Table 33-64.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | 0 | 1 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | SUBSTEP[3:0] | | | | YEAR[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 |
|---|---|---|---|---|---|---|---|---|
| | MON[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 |
|---|---|---|---|---|---|---|---|---|
| | DAY[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**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.

**Bits 19:16 – YEAR[3:0]** Timestamp Year
One digit, BCD-coded. This field is set by generic parameter on synthesis.

**Bits 15:8 – MON[7:0]** Timestamp Month
Two digits, BCD-coded. This field is set by generic parameter on synthesis.

**Bits 7:0 – DAY[7:0]** Timestamp Day
Two digits, BCD-coded. This field is set by generic parameter on synthesis.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 33.8.57. Timestamp Configuration

**Name:** TSCFG
**Offset:** 0x264
**Reset:** 0x00000000
**Property:** R/W

**Table 33-65.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | TBPRE[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SCP | TBCS | TSUE |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 15:8 – TBPRE[7:0]** Time Base Prescaler
0x00 to 0xFF
Valid values range from 0 to 255. The oscillator frequency is divided by TBPRE + 1 for generating the timebase counter clock.

**Bit 2 – SCP** Select Capturing Position

| Value | Description |
|---|---|
| 0 | Capture Timestamp at EOF |
| 1 | Capture Timestamp at SOF |

**Bit 1 – TBCS** Timebase Counter Select

| Value | Description |
|---|---|
| 0 | Timestamp value captured from internal timebase counter, ATB.TB[31:0] is the internal timebase counter. |
| 1 | Reserved |

**Bit 0 – TSUE** Timestamp Unit Enable

| Value | Description |
|---|---|
| 0 | TSU disabled |
| 1 | TSU enabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.58. Timestamp Status 1

**Name:** TSS1
**Offset:** 0x268
**Reset:** 0x00000000
**Property:** R

**Table 33-66.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TSN[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 |
|---|---|---|---|---|---|---|---|---|
| | TSN[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 |
|---|---|---|---|---|---|---|---|---|
| | TSL[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 |
|---|---|---|---|---|---|---|---|---|
| | TSL[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:16 – TSN[15:0]** Timestamp New
Each Timestamp register (TS0-TS7) is assigned one bit. The bits are set when a timestamp was stored in the related Timestamp register. Reading a Timestamp register resets the related bit.

**Bits 15:0 – TSL[15:0]** Timestamp Lost
Each Timestamp register (TS0-TS7) is assigned one bit. The bits are set when the timestamp stored in the related Timestamp register was overwritten before it was read. Reading a Timestamp register resets the related bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.59. Timestamp Status 2

**Name:** TSS2
**Offset:** 0x26C
**Reset:** 0x0000A000
**Property:** R

**Table 33-67.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | ITBG[1:0] | | NTSG[1:0] | | | | | |
| Access | R | R | R | R | | | | |
| Reset | 1 | 0 | 1 | 0 | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | TSP[3:0] | | | |
| Access | | | | | R | R | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bits 15:14 – ITBG[1:0]** Internal Timebase and SOF select Generic
This bitfield is fixed to 0x3.

| Value | Description |
|---|---|
| 00 | no SOF option, no internal timebase |
| 01 | no SOF option, internal timebase (default) |
| 10 | SOF option, no internal timebase |
| 11 | SOF option, internal timebase |

**Bits 13:12 – NTSG[1:0]** Number of Timestamps Generic
This parameter is fixed to 0x1.

| Value | Description |
|---|---|
| 00 | 4 timestamp registers (TS0-TS3) |
| 01 | 8 timestamp registers (default, TS0-TS7) |
| 10 | 16 timestamp registers (TS0-TS15) |
| 11 | Reserved |

**Bits 3:0 – TSP[3:0]** Timestamp Pointer
The Timestamp Pointer is incremented by one each time a timestamp is captured. From its maximum value (7), it is incremented to 0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 33.8.60. Timestamp 0

**Name:** TS0
**Offset:** 0x270
**Reset:** 0x00000000
**Property:** -

**Table 33-68.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS0


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.61. Timestamp 1

**Name:** TS1
**Offset:** 0x274
**Reset:** 0x00000000
**Property:** -

**Table 33-69.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS1


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.62. Timestamp 2

**Name:** TS2
**Offset:** 0x278
**Reset:** 0x00000000
**Property:** -

**Table 33-70.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.63. Timestamp 3

**Name:** TS3
**Offset:** 0x27C
**Reset:** 0x00000000
**Property:** -

**Table 33-71.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS0


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.64. Timestamp 4

**Name:** TS4
**Offset:** 0x280
**Reset:** 0x00000000
**Property:** -

**Table 33-72.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS4


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.65. Timestamp 5

**Name:** TS5
**Offset:** 0x284
**Reset:** 0x00000000
**Property:** -

**Table 33-73.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS5


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.66. Timestamp 6

**Name:** TS6
**Offset:** 0x288
**Reset:** 0x00000000
**Property:** -

**Table 33-74.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS6


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.67. Timestamp 7

**Name:** TS7
**Offset:** 0x28C
**Reset:** 0x00000000
**Property:** -

**Table 33-75.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS7


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.68. Timestamp 8

**Name:** TS8
**Offset:** 0x290
**Reset:** 0x00000000
**Property:** -

**Table 33-76.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS8


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.69. Timestamp 9

**Name:** TS9
**Offset:** 0x294
**Reset:** 0x00000000
**Property:** -

**Table 33-77.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS9


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.70. Timestamp 10

**Name:** TS10
**Offset:** 0x298
**Reset:** 0x00000000
**Property:** -

**Table 33-78.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS10


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.71. Timestamp 11

**Name:** TS11
**Offset:** 0x29C
**Reset:** 0x00000000
**Property:** -

**Table 33-79.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS11


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.72. Timestamp 12

**Name:** TS12
**Offset:** 0x2A0
**Reset:** 0x00000000
**Property:** -

**Table 33-80.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS12


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.73. Timestamp 13

**Name:** TS13
**Offset:** 0x2A4
**Reset:** 0x00000000
**Property:** -

**Table 33-81.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS13


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.74. Timestamp 14

**Name:** TS14
**Offset:** 0x2A8
**Reset:** 0x00000000
**Property:** -

**Table 33-82.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS14


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.75. Timestamp 15

**Name:** TS15
**Offset:** 0x2AC
**Reset:** 0x00000000
**Property:** -

**Table 33-83.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[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 |
|---|---|---|---|---|---|---|---|---|
| | TS[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TS[31:0]** Timestamp Word TS15


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.8.76. Actual Timebase

**Name:** ATB
**Offset:** 0x2B0
**Reset:** 0x00000000
**Property:** -

**Table 33-84.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TB[31:24] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | TB[23:16] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TB[15:8] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TB[7:0] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – TB[31:0]** Timebase for timestamp generation

## 33.9. Rx Buffer and FIFO Element Register Summary


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.9.1. Rx Buffer and FIFO Element 0

**Name:** R0
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** -

**Table 33-85.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ESI | XTD | RTR | ID[28:24] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 – ESI** Error State Indicator

| Value | Description |
|---|---|
| 0 | Transmitting node is error active. |
| 1 | Transmitting node is error passive. |

**Bit 30 – XTD** Extended Identifier
Signals to the Host whether the received frame has a standard or extended identifier

| Value | Description |
|---|---|
| 0 | 11-bit standard identifier. |
| 1 | 29-bit extended identifier |

**Bit 29 – RTR** Remote Transmission Request
Signals to the Host whether the received frame is a data frame or 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.

| Value | Description |
|---|---|
| 0 | Received frame is a data frame. |
| 1 | Received frame is a remote frame. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 28:0 – ID[28:0]** Identifier
Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.9.2. Rx Buffer and FIFO Element 1

**Name:** R1
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** RW

**Table 33-86.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ANMF | FIDX[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 |
|---|---|---|---|---|---|---|---|---|
| | | | FDF | BRS | DLC[3:0] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | RXTS[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 |
|---|---|---|---|---|---|---|---|---|
| | RXTS[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 – ANMF** Accepted Non-matching Frame
Acceptance of non-matching frames may be enabled via GFC.ANFS and GFC.ANFE

| Value | Description |
|---|---|
| 0 | Received frame matching filter index FIDX. |
| 1 | Received frame did not match any Rx filter element. |

**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.

**Bit 21 – FDF** FD Format

| Value | Description |
|---|---|
| 0 | Standard frame format. |
| 1 | CAN FD frame format (new DLC-coding and CRC). |

**Bit 20 – BRS** Bit Rate Search

| Value | Description |
|---|---|
| 0 | Frame received without bit rate switching. |
| 1 | Frame received with bit rate switching. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 19:16 – DLC[3:0]** Data Length Code

| Value | Description |
|---|---|
| 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 |

**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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.9.3. Rx Buffer and FIFO Element Data

**Name:** R2
**Offset:** 0x08 + n*0x04 [n=0..15]
**Reset:** 0x00000000
**Property:** RW

**Table 33-87.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DB3[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 |
|---|---|---|---|---|---|---|---|---|
| | DB2[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 |
|---|---|---|---|---|---|---|---|---|
| | DB1[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 |
|---|---|---|---|---|---|---|---|---|
| | DB0[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 – DB3[7:0]** Data Byte 3

**Bits 23:16 – DB2[7:0]** Data Byte 2

**Bits 15:8 – DB1[7:0]** Data Byte 1

**Bits 7:0 – DB0[7:0]** Data Byte 0

## 33.10. Tx Buffer Element Register Summary


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.10.1. Tx Buffer Element 0

**Name:** T0
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** -

**Table 33-88.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ESI | XTD | RTR | ID[28:24] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 – ESI** Error State Indicator

**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.

| Value | Description |
|---|---|
| 0 | ESI bit in CAN FD format depends only on error passive flag. |
| 1 | ESI bit in CAN FD format transmitted recessive. |

**Bit 30 – XTD** Extended Identifier

| Value | Description |
|---|---|
| 0 | 11-bit standard identifier. |
| 1 | 29-bit extended identifier. |

**Bit 29 – RTR** Remote Transmission Request

| Value | Description |
|---|---|
| 0 | Transmit data frame. |
| 1 | Transmit remote frame |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 28:0 – ID[28:0]** Identifier
Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.10.2. Tx Buffer Element 1

**Name:** T1
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** -

**Table 33-89.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | MML[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 |
|---|---|---|---|---|---|---|---|---|
| | EFC | TSCE | FDF | BRS | DLC[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 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | MMH[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bits 31:24 – MML[7:0]** Message Marker Lower Byte
Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

**Bit 23 – EFC** Event FIFO Control

| Value | Description |
|---|---|
| 0 | Don't store Tx events. |
| 1 | Store Tx events. |

**Bit 22 – TSCE** Time Stamp Capture Enable for TSU

**Bit 21 – FDF** FD Format

| Value | Description |
|---|---|
| 0 | Frame transmitted in Classic CAN format. |
| 1 | Frame transmitted in CAN FD format. |

**Bit 20 – BRS** Bit Rate Search
**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'.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | CAN FD frames transmitted without bit rate switching. |
| 1 | CAN FD frames transmitted with bit rate switching. |

**Bits 19:16 – DLC[3:0]** Identifier

| Value | Description |
|---|---|
| 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 |

**Bits 15:8 – MMH[7:0]** Message Marker Higher Byte


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.10.3. Tx Buffer Element Data

**Name:** T2
**Offset:** 0x08 + n*0x04 [n=0..15]
**Reset:** 0x00000000
**Property:** -

**Table 33-90.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | DB3[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 |
|---|---|---|---|---|---|---|---|---|
| | DB2[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 |
|---|---|---|---|---|---|---|---|---|
| | DB1[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 |
|---|---|---|---|---|---|---|---|---|
| | DB0[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 – DB3[7:0]** Data Byte 3

**Bits 23:16 – DB2[7:0]** Data Byte 2

**Bits 15:8 – DB1[7:0]** Data Byte 1

**Bits 7:0 – DB0[7:0]** Data Byte 0

## 33.11. Tx Event FIFO Element Register Summary


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.11.1. Tx Event FIFO Element 0

**Name:** E0
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** -

**Table 33-91.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | ESI | XTD | RTR | ID[28:24] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 |
|---|---|---|---|---|---|---|---|---|
| | ID[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 – ESI** Error State Indicator

| Value | Description |
|---|---|
| 0 | Transmitting node is error active. |
| 1 | Transmitting node is error passive. |

**Bit 30 – XTD** Extended Indentifier

| Value | Description |
|---|---|
| 0 | 11-bit standard identifier |
| 1 | 29-bit extended identifier |

**Bit 29 – RTR** Remote Transmission Request

| Value | Description |
|---|---|
| 0 | Received frame is a data frame. |
| 1 | Received frame is a remote frame. |

**Bits 28:0 – ID[28:0]** Identifier
Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.11.2. Tx Event FIFO Element 1

**Name:** E1
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** RW

**Table 33-92.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | MM[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 |
|---|---|---|---|---|---|---|---|---|
| | ET[1:0] | | FDF | BRS | DLC[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 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TXTS[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 |
|---|---|---|---|---|---|---|---|---|
| | TXTS[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 – MM[7:0]** Message Marker
Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status

**Bits 23:22 – ET[1:0]** Event Type
This field defines the 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) |

**Bit 21 – FDF** FD Format

| Value | Description |
|---|---|
| 0 | Standard frame format. |
| 1 | CAN FD frame format (new DLC-coding and CRC). |

**Bit 20 – BRS** Bit Rate Search

| Value | Description |
|---|---|
| 0 | Frame received without bit rate switching. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 1 | Frame received with bit rate switching. |

**Bits 19:16 – DLC[3:0]** Data Length Code

| Value | Description |
|---|---|
| 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 |

**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.

## 33.12. Standard Message ID Filter Element Register Summary


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.12.1. Standard Message ID Filter Element

**Name:** S0
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** -

**Table 33-93.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SFT[1:0] | | SFEC[2:0] | | | SFID1[10:8] | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | SFID1[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | SFID2[10:8] | | |
| Access | | | | | | R/W | R/W | R/W |
| Reset | | | | | | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SFID2[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 31:30 – SFT[1:0]** Standard Filter Type
This field defines the standard filter type

| Value | Name | Description |
|---|---|---|
| 0 | RANGE | Range filter from SFID1 to SFID2 |
| 1 | DUAL | Dual ID filter for SFID1 or SFID2 |
| 2 | CLASSIC | Classic filter |

**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.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Disable filter element |
| 1 | STF0M | Store in Rx FIFO 0 if filter match |
| 2 | STF1M | Store in Rx FIFO 1 if filter match |
| 3 | REJECT | Reject ID if filter match |
| 4 | PRIORITY | Set priority if filter match |
| 5 | PRIF0M | Set priority and store in FIFO 0 if filter match |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 6 | PRIF1M | Set priority and store in FIFO 1 if filter match |
| 7 | STRXBUF | Store into Rx Buffer |

**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 10:0 – SFID2[10:0]** Standard Filter ID 2
This bit field has a different meaning depending on the configuration of SFEC.
1. SFEC = “001” ... “110”: Second ID of standard ID filter element.
2. 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
01 = Debug Message A
10 = Debug Message B
11 = Debug Message C
SFID2[8:6] is reserved for future usage.
SFID2[5:0] defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message

## 33.13. Extended Message ID Filter Element Register Summary


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.13.1. Extended Message ID Filter Element 0

**Name:** F0
**Offset:** 0x000
**Reset:** 0x00000000
**Property:** -

**Table 33-94.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | EFEC[2:0] | | | EFID1[28:24] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | EFID1[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 |
|---|---|---|---|---|---|---|---|---|
| | EFID1[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 |
|---|---|---|---|---|---|---|---|---|
| | EFID1[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 31:29 – 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 register HPMS is updated with the status of the priority match.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Disable filter element |
| 1 | STF0M | Store in Rx FIFO 0 if filter match |
| 2 | STF1M | Store in Rx FIFO 1 if filter match |
| 3 | REJECT | Reject ID if filter match |
| 4 | PRIORITY | Set priority if filter match |
| 5 | PRIF0M | Set priority and store in FIFO 0 if filter match |
| 6 | PRIF1M | Set priority and store in FIFO 1 if filter match |
| 7 | STRXBUF | Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored |

**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 mesage to be stored. The received identifiers must match exactly, only XIDAM masking mechanism is used.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 33.13.2. Extended Message ID Filter Element 1

**Name:** F1
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** -

**Table 33-95.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | EFT[1:0] | | | EFID2[28:24] | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | EFID2[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 |
|---|---|---|---|---|---|---|---|---|
| | EFID2[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 |
|---|---|---|---|---|---|---|---|---|
| | EFID2[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 31:30 – EFT[1:0]** Extended Filter Type
This field defines the extended filter type.

| Value | Name | Description |
|---|---|---|
| 0 | RANGEM | Range filter from EFID1 to EFID2 |
| 1 | DUAL | Dual ID filter for EFID1 or EFID2 |
| 2 | CLASSIC | Classic filter |
| 3 | RANGE | Range filter from EFID1 to EFID2 with no XIDAM mask |

**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 reserved for future usage. 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.

# 34. Universal Serial Bus (USB)

## 34.1. Overview

The Universal Serial Bus interface (USB) module complies with the Universal Serial Bus (USB) 2.1 specification supporting device 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 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.

## 34.2. Features

The following are key features of the USB module:

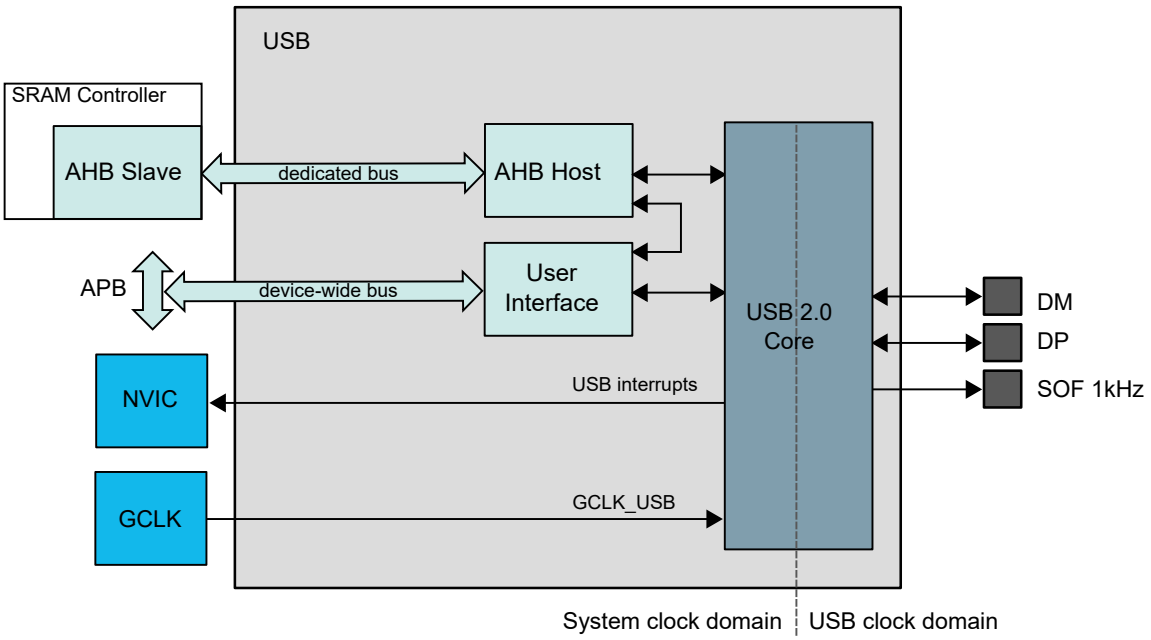
- Compatible with the USB 2.1 specification
- USB Host and Device mode
- Supports full (12 Mbit/s) and low (1.5 Mbit/s) speed communication
- Supports Link Power Management (LPM-L1) protocol
- On-chip transceivers with built-in pull-ups and pull-downs
- On-Chip USB serial resistors
- 1 kHz SOF clock 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Built-in DMA with multi-packet support and dual bank for all pipes
- Supports feedback endpoint
- Supports the USB 2.0 Phase-locked SOFs feature

## 34.3. USB Block Diagram

**Figure 34-1.** LS/FS Implementation: USB Block Diagram




## 34.4. 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 |

**Note:** Refer to I/O Multiplexing and Considerations for details on the pin mapping for this peripheral. One signal can be mapped on several pins..

## 34.5. Product Dependancies

In order to use this peripheral, other parts of the system must be configured correctly, as described below.

### 34.5.1. 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 1 kHz SOF clock is available on an external pin. The user must first configure the I/O Controller to assign the 1 kHz SOF clock to the peripheral function. The SOF clock is available for device and host mode.

### 34.5.2. Power Managemnet

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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.5.3. 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.

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.

### 34.5.4. 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.

### 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.

### 34.5.6. 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.

### 34.5.7. Register Access Protection

Registers with write access can be optionally write-protected by the Peripheral Access Controller (PAC), except for the following:

- The Device Interrupt Flag (INTFLAG) register
- The Endpoint Interrupt Flag (EPINTFLAG) register
- The Host Interrupt Flag (INTFLAG) register
- The 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.

### 34.5.8. 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.

## 34.6. Functional Description

### 34.6.1. USB General Operation

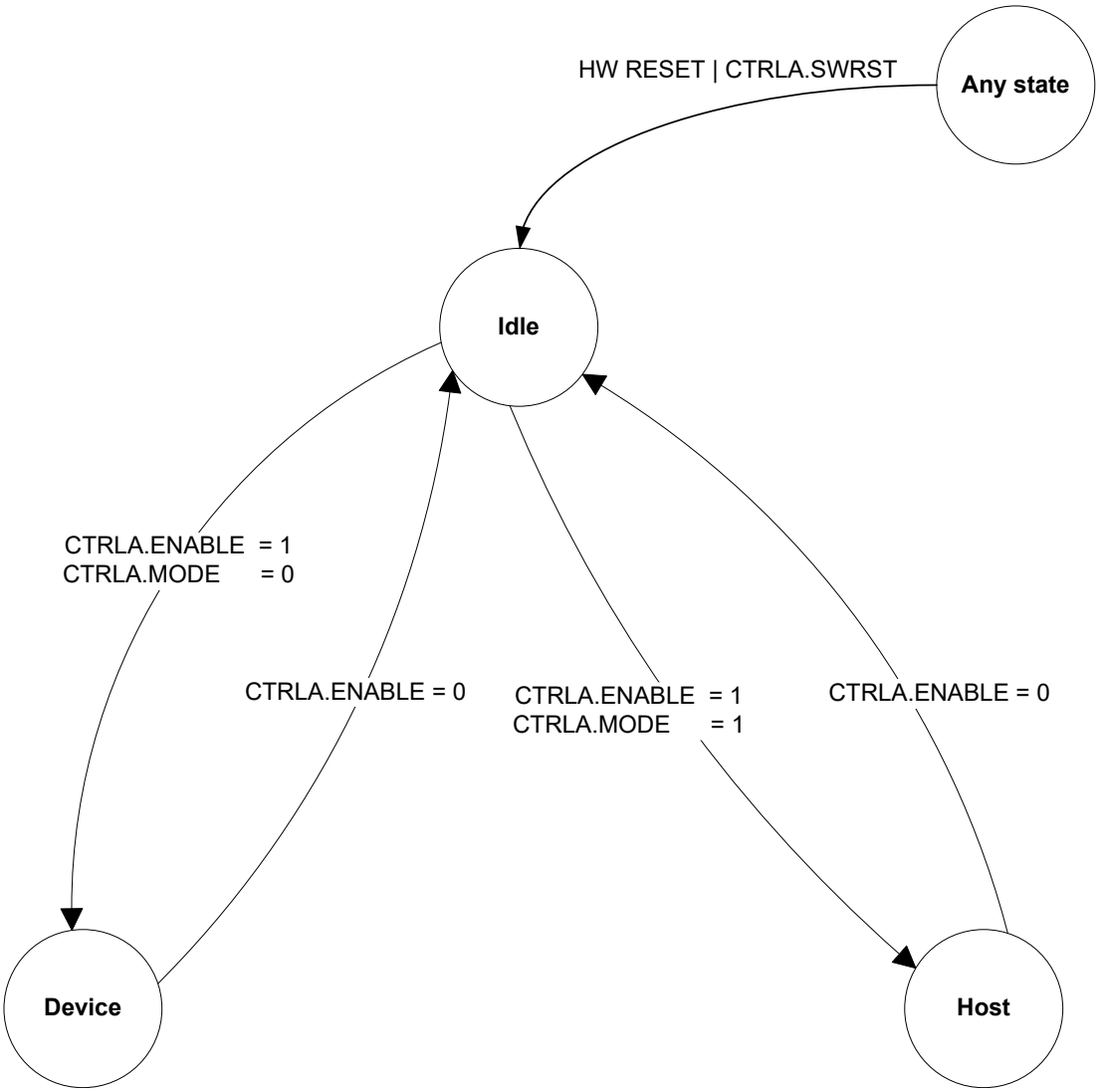
#### 34.6.1.1. Initialization

After a hardware reset, the USB is in the idle state. In this state:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- 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 are reset.

**Figure 34-2.** General States




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 Control A (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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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.

Refer to USB Device Operations for the basic operation of the device mode.

### 34.6.2. 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 2.1.

#### 34.6.2.1. 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.

#### 34.6.2.2. 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. Refer to the section *Endpoint Descriptor 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:

- The Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- The Device Endpoint Interrupt Flag (EPINTFLAG) register
- The Transmit Stall '0' bit in the Endpoint Status (EPSTATUS.STALLRQ0) register
- The Transmit Stall '1' bit in the Endpoint Status (EPSTATUS.STALLRQ1) register

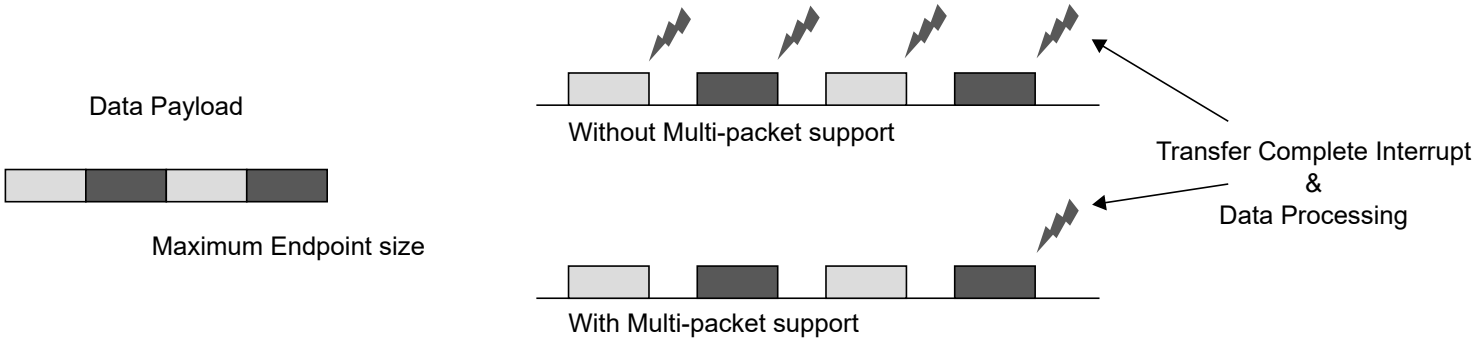
#### 34.6.2.3. Multi-Packet Transfers

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.

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 34-3.** Multi-Packet Feature - Reduction of CPU Overhead




#### 34.6.2.4. USB Reset

The USB bus reset is initiated by a connected host and managed by hardware.

During USB reset the following registers are cleared:

- The Device Endpoint Configuration (EPCFG) register - except for Endpoint 0
- The Device Frame Number (FNUM) register
- The Device Address (DADD) register
- The Device Endpoint Interrupt Enable Clear/Set (EPINTENCLR/SET) register
- The Device Endpoint Interrupt Flag (EPINTFLAG) register
- The Transmit Stall '0' bit in the Endpoint Status (EPSTATUS.STALLRQ0) register
- The Transmit Stall '1' bit in the Endpoint Status (EPSTATUS.STALLRQ1) register
- The Endpoint Interrupt Summary (EPINTSMRY) register
- The Upstream resume bit in the Control B (CTRLB.UPRSM) register

At the end of the reset process, the End of Reset bit is set in the Interrupt Flag register (INTFLAG.EORST).

#### 34.6.2.5. 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.

#### 34.6.2.6. 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.

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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

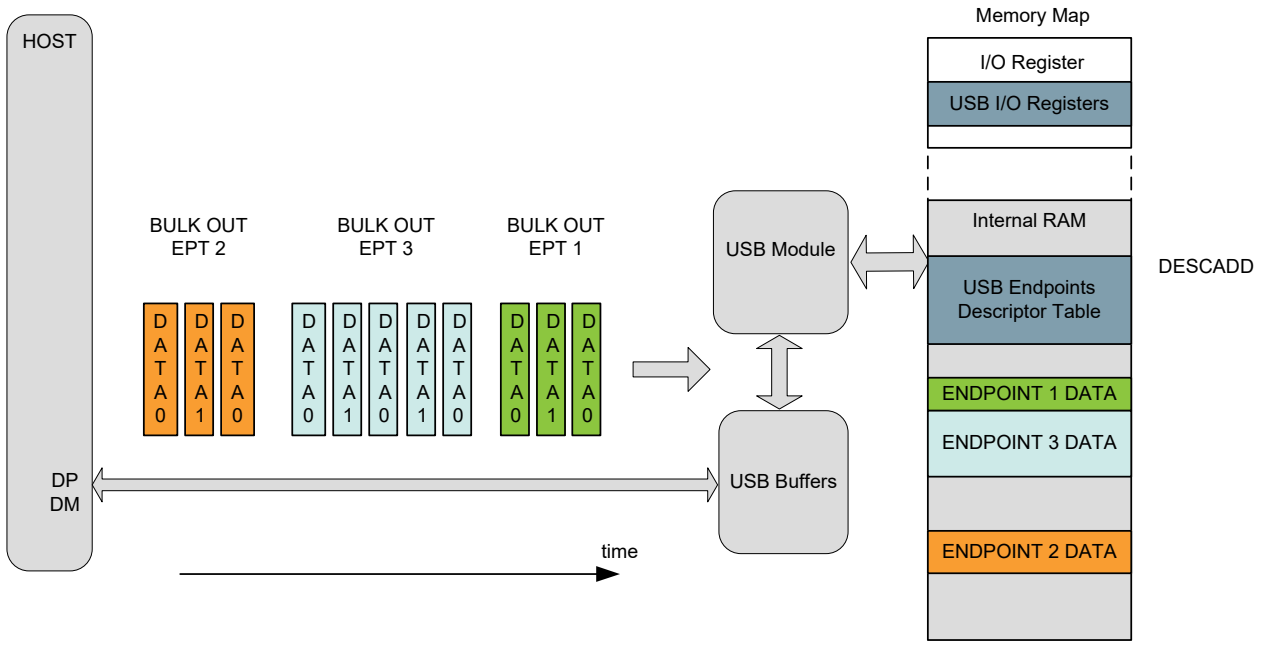
Device Endpoint Status register (EPSTATUS.BK0RDY), the incoming data is written to the data buffer pointed to by the Data Buffer Address (ADDR). If the number of received data bytes exceeds the endpoint's maximum data payload size as specified by the PCKSIZE.SIZE, the remainders of the received data 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. The 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. The 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 the Endpoint Interrupt Enable Set/Clear register (EPINTENSET/CLR.RXSTP).

### 34.6.2.7. Management of OUT Transactions

**Figure 34-4.** 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 the EPINTFLAG (EPINTFLAG.STALL0) register 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.

#### 34.6.2.8. 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.

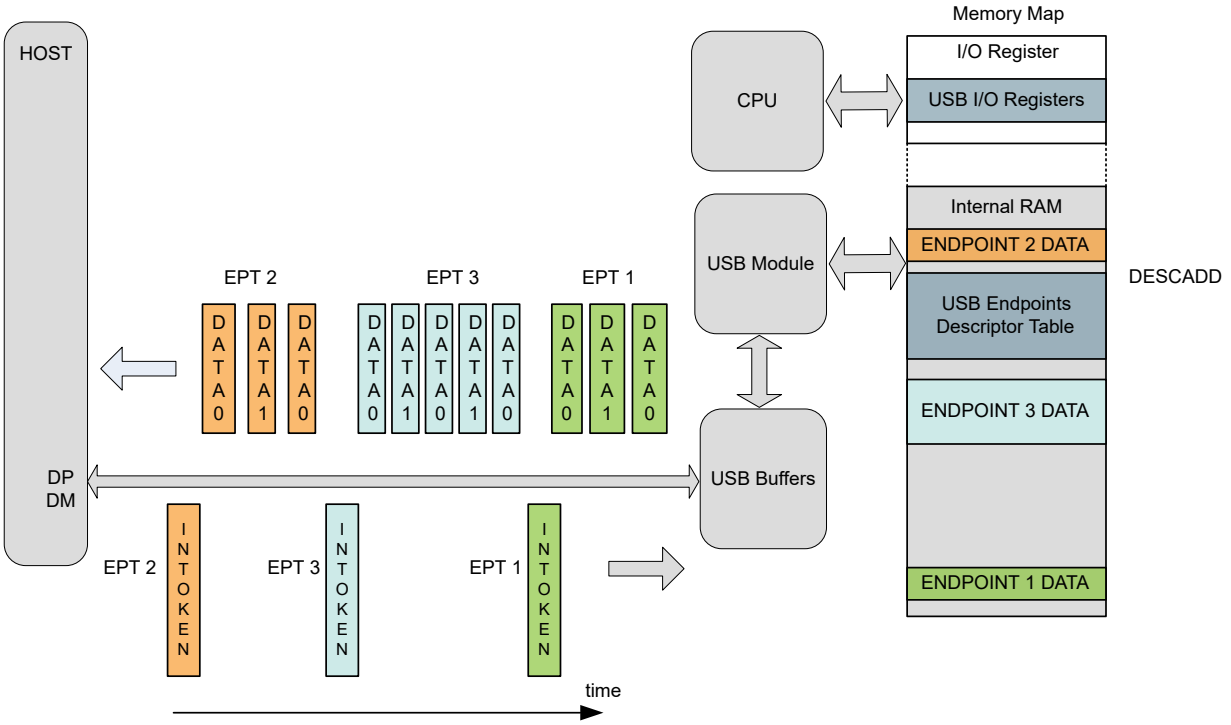
EPSTATUS.DTGLOUT management for non-isochronous packets and EPINTFLAG.BK1RDY/BK0RDY management are as for normal operation.

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.

### 34.6.2.9. Management of IN Transactions

**Figure 34-5.** 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.

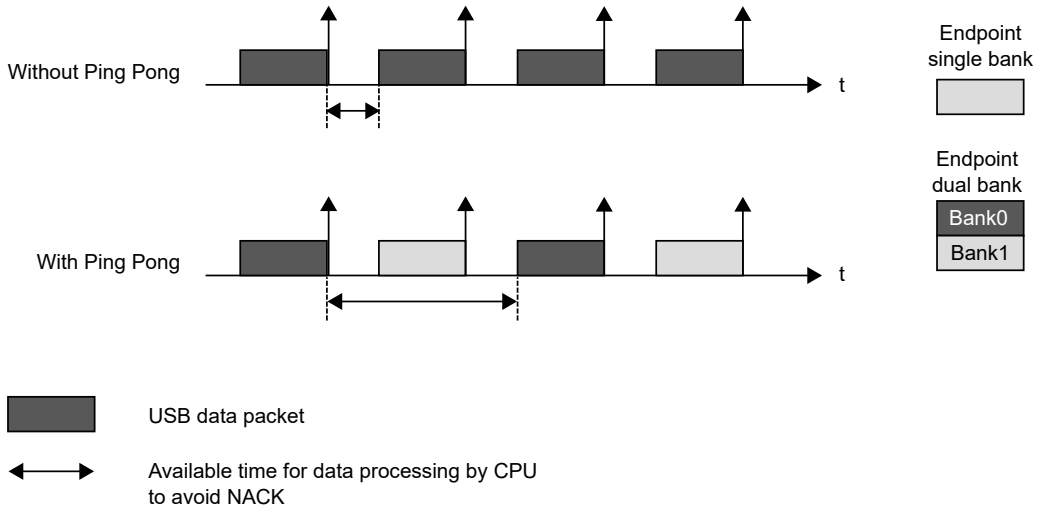
For all non-isochronous endpoints the USB module waits for an ACK handshake from the host. If an ACK handshake is not received within 16 bit times, the USB module returns to idle and waits for the next token packet. If an ACK handshake is successfully received EPSTATUS.BK1RDY is cleared, EPINTFLAG.TRCPT1 is set and EPSTATUS.DTGLIN is toggled.

### 34.6.2.10. 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 34-6.** 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 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.

### 34.6.2.11. Feedback Operation

Feedback endpoints are endpoints with the same 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 endpoints. 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:

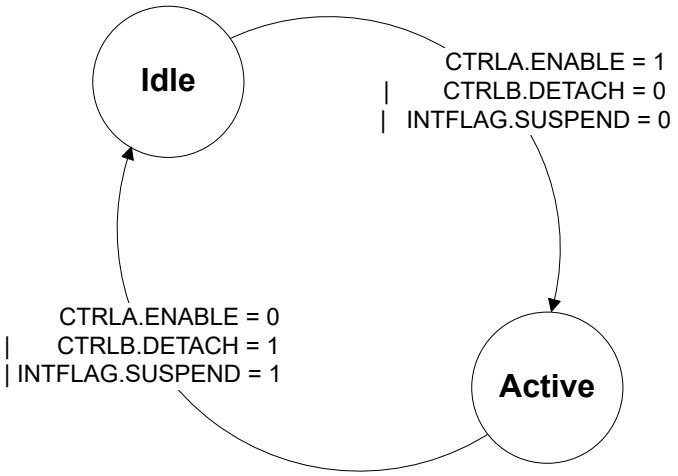
- Endpoint n / IN: EPCFG.EPTYPE1 = Interrupt IN, PCKSIZE.SIZE = 64.
- Endpoint n / OUT: EPCFG.EPTYPE0 = Isochronous OUT, PCKSIZE.SIZE = 512.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.6.2.12. Suspend State and Pad Behavior

The following figure illustrates the behavior of the USB pad in Device mode.

**Figure 34-7.** Pad Behavior

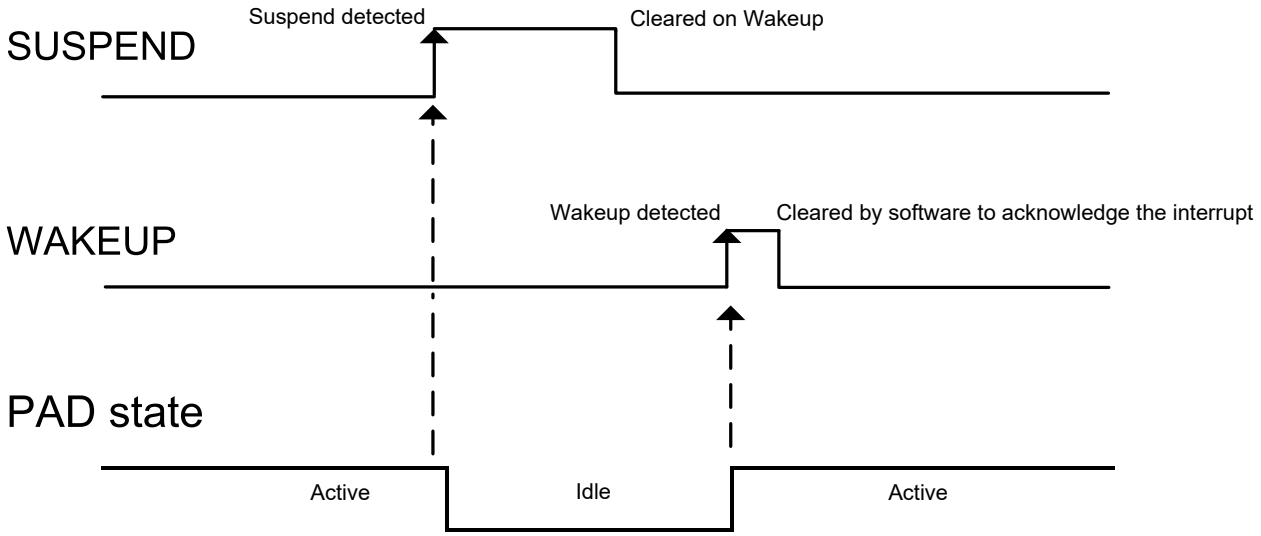



In Idle state, the pad is in Low-Power Consumption mode.

In Active state, the pad is active.

The following figure illustrates the pad events leading to a PAD state change.

**Figure 34-8.** Pad Events





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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.

#### 34.6.2.13. 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, that is, 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.

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.

#### 34.6.2.14. 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) 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 (see the BESL parameter in the Extended Register (EXTREG)).

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.

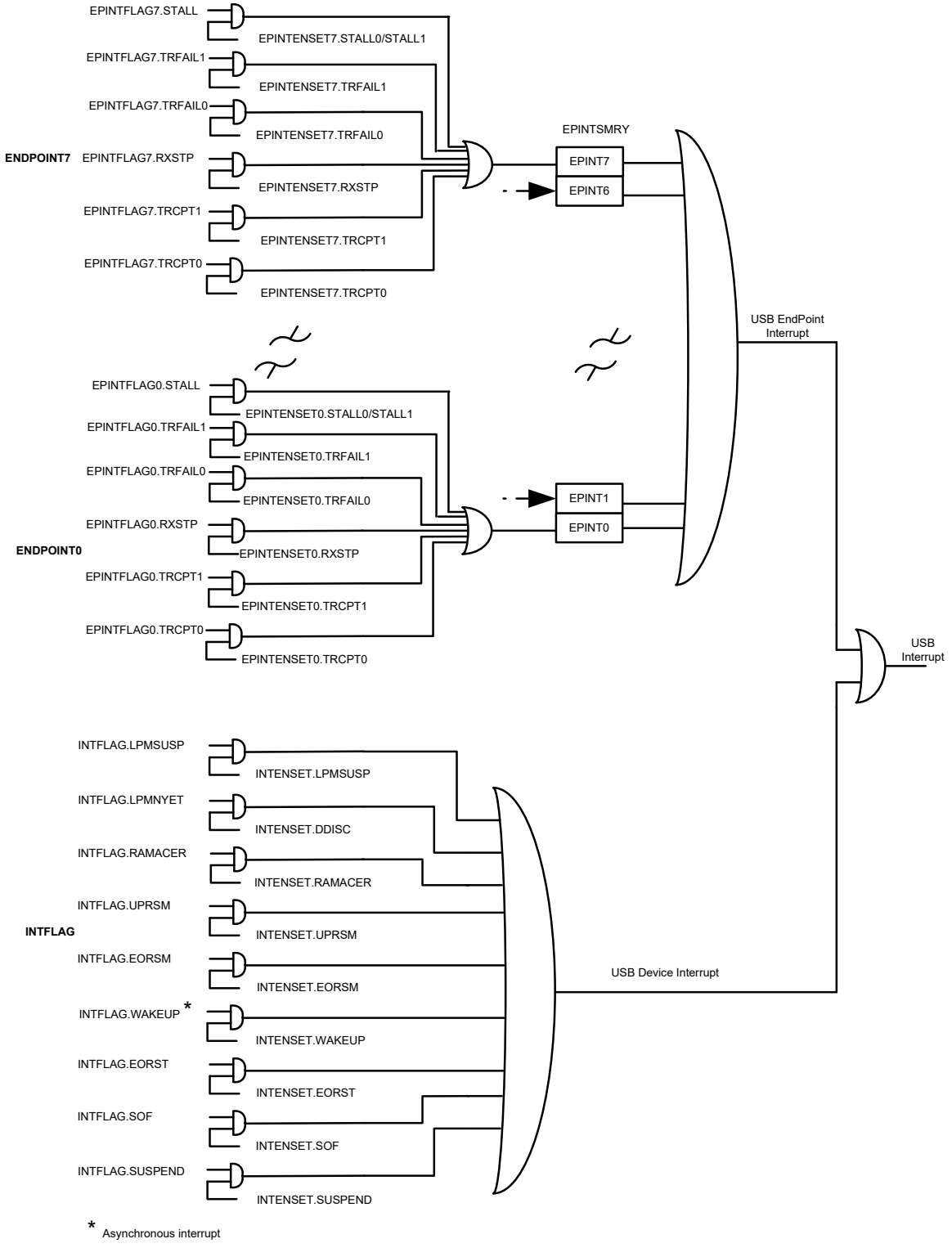
If the LPM transaction is responded with a NYET, the Link Power Management Not Yet Interrupt Flag (INTFLAG.LPMNYET) is set. This generates an interrupt if the Link Power Management Not Yet Interrupt Enable bit (INTENCLR/SET.LPMNYET) is set.

If the LPM transaction is responded with no handshake, no flag is set, and the transaction is ignored.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 34.6.2.15. USB Device Interrupt

**Figure 34-9.** 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.

### 34.6.3. 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.

#### 34.6.3.1. Device Detection and Disconnection

Prior to device detection, the software must set the VBUS is OK bit (CTRLB.VBUSOK) in the CTRLB register when the VBUS is available. This notifies the USB host that USB operations can be started. When the CTRLB.VBUSOK bit is '0' and even if the USB HOST is configured and enabled, host operation is halted. Setting the CTRLB.VBUSOK bit 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.

#### 34.6.3.2. Host Termin

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.

#### 34.6.3.3. 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:

- The All Host Pipe Configuration register (PCFG)
- The Host Frame Number register (FNUM)
- The Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- The Host Start-of-Frame Control register (HSOFC)
- The Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- The Pipe Interrupt Flag register (PINTFLAG)
- The 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.

#### 34.6.3.4. 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 the section *Pipe Descriptor 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 the 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 '1' 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:

- The Interval for the Bulk-Out/Ping transaction register (BINTERVAL)
- The Pipe Interrupt Enable Clear/Set register (PINTENCLR/SET)
- The Pipe Interrupt Flag register (PINTFLAG)
- The Pipe Freeze bit in Pipe Status register (PSTATUS.FREEZE)

#### 34.6.3.5. 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.

#### 34.6.3.6. 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.

#### 34.6.3.7. 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 the INTFLAG register (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 the CTRLB register (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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Resume on the 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.

#### 34.6.3.8. 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, and 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, and 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.

#### 34.6.3.9. 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

#### 34.6.3.10. 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.

#### 34.6.3.11. 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.

#### 34.6.3.12. 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.

#### 34.6.3.13. 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.

#### 34.6.3.14. 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.

#### 34.6.3.15. 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

maximum allowed number of Pipe error(s) 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.

#### 34.6.3.16. 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 Extended Register (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. 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. See Extended Register (EXTREG) register.

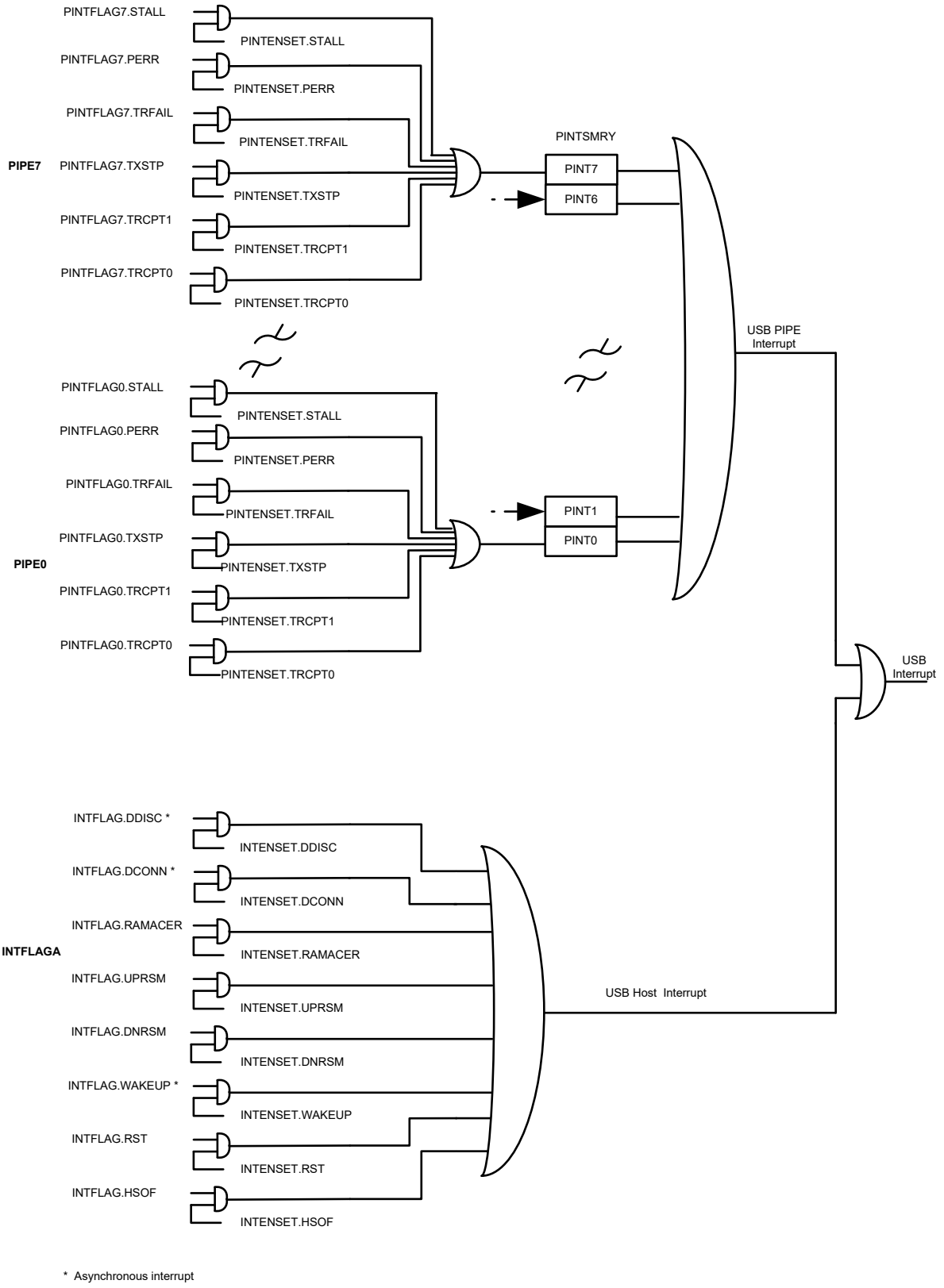
When the host is in the L1 SLEEP state after a successful LPM transmitted, the device can initiate an UPSTREAM RESUME. This will set the Upstream Resume Interrupt bit in INTFLAG (INTFLAG.UPRSM). The host should proceed then to a L1 RESUME as described above.

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.

### 34.6.3.17. Host Interrupt

**Figure 34-10.** 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.

## 34.7. Peripheral Dependencies

**Table 34-1.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| USB | 0x44832000 | 134 : EORSMDNRSM/ EORSTRST/ LPMDCONN/ LPMSUSPDDISC/ MSOF/ RAMACER/ RXSTPTXSTP0/ RXSTPTXSTP1/ RXSTPTXSTP2/ RXSTPTXSTP3/ RXSTPTXSTP4/ RXSTPTXSTP5/ RXSTPTXSTP6/ RXSTPTXSTP7/ STALL0STALL0/ STALL0STALL1/ STALL0STALL2/ STALL0STALL3/ STALL0STALL4/ STALL0STALL5/ STALL0STALL6/ STALL0STALL7/ STALL10/ STALL11/ STALL12/ STALL13/ STALL14/ STALL15/ STALL16/ STALL17/ SUSPEND/ TRFAIL0TRFAIL0/ TRFAIL0TRFAIL1/ TRFAIL0TRFAIL2/ TRFAIL0TRFAIL3/ TRFAIL0TRFAIL4/ TRFAIL0TRFAIL5/ TRFAIL0TRFAIL6/ TRFAIL0TRFAIL7/ TRFAIL1PERR0/ TRFAIL1PERR1/ TRFAIL1PERR2/ TRFAIL1PERR3/ TRFAIL1PERR4/ TRFAIL1PERR5/ TRFAIL1PERR6/ TRFAIL1PERR7/ UPRSM/ WAKEUP<br>135 : SOFHSOF<br>136 : TRCPT00/ TRCPT01/ TRCPT02/ TRCPT03/ TRCPT04/ TRCPT05/ TRCPT06/ TRCPT07<br>137 : TRCPT10/ TRCPT11/ TRCPT12/ TRCPT13/ TRCPT14/ TRCPT15/ TRCPT16/ TRCPT17 | **AHB:** CLKMSK[1] MASK22 **APB:** CLKMSK[1] MASK23 | PCHCTRL[38] | 44 STATUS1 PERID12 | | | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 34.8. 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.

### 34.8.1. Control A

**Name:** CTRLA
**Offset:** 0x000
**Reset:** 0x00
**Property:** PAC Write-Protection, Write-Synchronised

**Table 34-2.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 ModeThis bit defines the operating mode of the USB.

| Value | Name | Description |
|---|---|---|
| 0 | DEVICE | Device Mode |
| 1 | HOST | 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.
This bit is Write-Synchronized.

| 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. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x002
**Reset:** 0x00
**Property:** R

**Table 34-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.8.3. USB Quality Of Service

**Name:** QOSCTRL
**Offset:** 0x03
**Reset:** 0x0F
**Property:** PAC Write-Protection

**Table 34-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | | | 1 | 1 | 1 | 1 |

**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.

### 34.8.4. Finite State Machine Status

**Name:** FSMSTATUS
**Offset:** 0x00D
**Reset:** 0x01
**Property:** R

**Table 34-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.8.5. Descriptor Address

**Name:** DESCADD
**Offset:** 0x024
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 34-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Register
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.

### 34.8.6. USB PAD Calibration

**Name:** PADCAL
**Offset:** 0x028
**Reset:** 0x0000
**Property:** PAC Write-Protection

**Table 34-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

## 34.9. Register Summary - DEVICE

| 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] | | | | | MFNUM[2:0] | | |
| | | 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 | EPINT[7:0] | | | | | | | |
| | | 15:8 | EPINT[15:8] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.9.1. Control B

**Name:** CTRLB
**Offset:** 0x008
**Reset:** 0x0000
**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 | 0 |

**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.

### 34.9.2. 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.

### 34.9.3. 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.

### 34.9.4. 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] | | | | | MFNUM[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 – FNCERR** Frame Number CRC Error
This bit is cleared upon receiving a USB reset.
This bit is set when a corrupted frame number is received.
This bit and the SOF 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.

**Bits 2:0 – MFNUM[2:0]** Micro Frame Number
These bits are cleared upon receiving a USB reset or at the beginning of each Start-of-Frame (SOF interrupt).
These bits are updated with the micro-frame number information as provided from the last MSOF packet even if a corrupted MSOF is received.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.9.5. 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.

### 34.9.6. 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.

### 34.9.7. 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,.
In High Speed mode, the MFNUM register is cleared.
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.

### 34.9.8. End Point Interrupt Summary

**Name:** EPINTSMRY
**Offset:** 0x020
**Reset:** 0x0000
**Property:** -

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | EPINT[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 |
|---|---|---|---|---|---|---|---|---|
| | EPINT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – EPINT[15:0]** End Point Interrupt
The flag EPINT[n] is set when an interrupt is triggered by the EndPoint n. See EPINTFLAG 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.

## 34.10. Device Endpoint 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 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0108 | EPINTENCLR0 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0109 | EPINTENSET0 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x010A ... 0x0113 | Reserved | | | | | | | | | |
| 0x0114 | EPCFG1 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0115 ... 0x0117 | Reserved | | | | | | | | | |
| 0x0118 | EPSTATUSCLR1 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0119 | EPSTATUSSET1 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x011A | EPSTATUS1 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x011B | EPINTFLAG1 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x011C | EPINTENCLR1 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x011D | EPINTENSET1 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x011E ... 0x0127 | Reserved | | | | | | | | | |
| 0x0128 | EPCFG2 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0129 ... 0x012B | Reserved | | | | | | | | | |
| 0x012C | EPSTATUSCLR2 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x012D | EPSTATUSSET2 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x012E | EPSTATUS2 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x012F | EPINTFLAG2 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0130 | EPINTENCLR2 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0131 | EPINTENSET2 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0132 ... 0x013B | Reserved | | | | | | | | | |
| 0x013C | EPCFG3 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x013D ... 0x013F | Reserved | | | | | | | | | |
| 0x0140 | EPSTATUSCLR3 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0141 | EPSTATUSSET3 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0142 | EPSTATUS3 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0143 | EPINTFLAG3 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0144 | EPINTENCLR3 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0145 | EPINTENSET3 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0146 ... 0x014F | Reserved | | | | | | | | | |
| 0x0150 | EPCFG4 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0151 ... 0x0153 | Reserved | | | | | | | | | |
| 0x0154 | EPSTATUSCLR4 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Device Endpoint Register Summary** (continued)

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0155 | EPSTATUSSET4 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0156 | EPSTATUS4 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0157 | EPINTFLAG4 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0158 | EPINTENCLR4 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0159 | EPINTENSET4 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x015A ... 0x0163 | Reserved | | | | | | | | | |
| 0x0164 | EPCFG5 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0165 ... 0x0167 | Reserved | | | | | | | | | |
| 0x0168 | EPSTATUSCLR5 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0169 | EPSTATUSSET5 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x016A | EPSTATUS5 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x016B | EPINTFLAG5 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x016C | EPINTENCLR5 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x016D | EPINTENSET5 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x016E ... 0x0177 | Reserved | | | | | | | | | |
| 0x0178 | EPCFG6 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x0179 ... 0x017B | Reserved | | | | | | | | | |
| 0x017C | EPSTATUSCLR6 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x017D | EPSTATUSSET6 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x017E | EPSTATUS6 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x017F | EPINTFLAG6 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0180 | EPINTENCLR6 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0181 | EPINTENSET6 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0182 ... 0x018B | Reserved | | | | | | | | | |
| 0x018C | EPCFG7 | 7:0 | | | EPTYPE1[2:0] | | | | EPTYPE0[2:0] | |
| 0x018D ... 0x018F | Reserved | | | | | | | | | |
| 0x0190 | EPSTATUSCLR7 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0191 | EPSTATUSSET7 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0192 | EPSTATUS7 | 7:0 | BK1RDY | BK0RDY | STALLRQ1 | STALLRQ0 | | CURBK | DTGLIN | DTGLOUT |
| 0x0193 | EPINTFLAG7 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0194 | EPINTENCLR7 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| 0x0195 | EPINTENSET7 | 7:0 | | | STALL | RXSTP | | TRFAIL | | TRCPT |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.10.1. Device Endpoint Configuration register n

**Name:** EPCFGn
**Offset:** 0x0100 + n*0x14 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 34-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.10.2. EndPoint Status Clear n

**Name:** EPSTATUSCLRn
**Offset:** 0x0104 + n*0x14 [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.

### 34.10.3. End Point Pipe Status Set

**Name:** EPSTATUSSETn
**Offset:** 0x0105 + n*0x14 [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.

### 34.10.4. End Point Pipe Status n

**Name:** EPSTATUSn
**Offset:** 0x0106 + n*0x14 [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 | 2 | 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. For Control/OUT direction Endpoints, the bank is empty. |
| 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. |

**Bit 5 – STALLRQ1** STALL bank 1 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 STALLRQ1 feature. |
| 1 | Enable STALLRQ1 feature: a STALL handshake will be sent to the host in regards to bank 1. |

**Bit 4 – STALLRQ0** STALL bank 0 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 STALLRQ0 feature. |
| 1 | Enable STALLRQ0 feature: a STALL handshake will be sent to the host in regards to bank 0. |

**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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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. |

**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.

### 34.10.5. End Point Interrupt Flag n

**Name:** EPINTFLAGn
**Offset:** 0x0107 + n*0x14 [n=0..7]
**Reset:** 0x00
**Property:** -

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| Access | | | R/W | R/W | | R/W | | R/W |
| Reset | | | 0 | 0 | | 0 | | 0 |

**Bit 5 – 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.

**Bit 2 – 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.

**Bit 0 – 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.

### 34.10.6. End Point Interrupt Clear Flag

**Name:** EPINTENCLRn
**Offset:** 0x0108 + n*0x14 [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 |
|---|---|---|---|---|---|---|---|---|
| | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| Access | | | R/W | R/W | | R/W | | R/W |
| Reset | | | 0 | 0 | | 0 | | 0 |

**Bit 5 – 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. |

**Bit 2 – 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. |

**Bit 0 – 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.

### 34.10.7. End Point Interrupt Set Flag

**Name:** EPINTENSETn
**Offset:** 0x0109 + n*0x14 [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 or when EPEN[n] is zero.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | STALL | RXSTP | | TRFAIL | | TRCPT |
| Access | | | R/W | R/W | | R/W | | R/W |
| Reset | | | 0 | 0 | | 0 | | 0 |

**Bit 5 – 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 Stall x 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. |

**Bit 2 – 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. |

**Bit 0 – 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 bank 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. |

## 34.11. USB Descriptor Register Summary


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.11.1. Endpoint Bank, Address of Data Buffer

**Name:** ADDR - Device
**Offset:** 0x00 + evice*0x04 [evice=0..1]
**Reset:** 0xxxxxxxx
**Property:** NA

**Table 34-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.11.2. Device Endpoint Bank, Packet Size

**Name:** PCKSIZE
**Offset:** 0x04 + n*0x04 [n=0..1]
**Reset:** 0xxxxxxxxx
**Property:** NA

**Table 34-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:**

1. For Isochronous endpoints 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.
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.

### 34.11.3. Endpoint Bank, Extended

**Name:** EXTREG - Device
**Offset:** 0x08
**Reset:** 0xxxxxxxx
**Property:** NA

**Table 34-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.11.4. Enpoint Bank, Status of Bank

**Name:** STATUS_BK
**Offset:** 0x0A + n*0x0A [n=0..1]
**Reset:** 0xxxxxxxx
**Property:** NA

**Table 34-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

| Value | Description |
|---|---|
| 0 | No CRC Error. |
| 1 | CRC Error detected. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 34.12. Register Summary - HOST

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 ... 0x07 | Reserved | | | | | | | | | |
| 0x08 | CTRLB | 7:0 | | | | | 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 | PINT[7:0] | | | | | | | |
| | | 15:8 | PINT[15:8] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.12.1. 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | SPDCONF[1:0] | | RESUME | |
| Access | | | | | R/W | R/W | R/W | |
| Reset | | | | | 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 also 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. |

**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 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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.

### 34.12.2. Host Start Of Frame Control

**Name:** HSOFC
**Offset:** 0x00A
**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.

### 34.12.3. Status

**Name:** STATUS
**Offset:** 0x00C
**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.

### 34.12.4. 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.
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.

### 34.12.5. Host Frame Length

**Name:** FLENHIGH
**Offset:** 0x012
**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 34-13.** Counter Description vs. Speed

| Host Register STATUS.SPEED | Description |
|---|---|
| Full Speed | With a USB clock running at 12MHz, counter length is 12000 to ensure a SOF generation every 1 ms. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.12.6. Host 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 Pipe Interrupt Enable Set (PINTENSET) 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.

### 34.12.7. Host 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 Pipe Interrupt Enable Set (PINTENCLR) 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.

### 34.12.8. Host Interrupt Flag

**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.

### 34.12.9. Pipe Interrupt Summary

**Name:** PINTSMRY
**Offset:** 0x020
**Reset:** 0x0000
**Property:** Read-only

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | PINT[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 |
|---|---|---|---|---|---|---|---|---|
| | PINT[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 15:0 – PINT[15:0]**

The flag PINT[n] is set when an interrupt is triggered by the pipe n. See 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.

## 34.13. Host Registers - Pipe - 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 | BINTERVAL - HOST_PIPE Mode0 | 7:0 | BINTERVAL[7:0] | | | | | | | |
| 0x0104 | PSTATUSCLR0 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0105 | PSTATUSSET0 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0106 | PSTATUS - HOST_PIPE Mode0 | 7:0 | BK1RDY | BK0RDY | | PFREEZE | | CURBK | | DTGL |
| 0x0107 | PINTFLAG0 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | | TRCPT |
| 0x0108 | PINTENCLR0 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | | TRCPT |
| 0x0109 | PINTENSET0 | 7:0 | | | STALL | TXSTP | PERR | TRFAIL | | TRCPT |
| 0x010A ... 0x0113 | Reserved | | | | | | | | | |
| 0x0114 | PCFG1 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0115 ... 0x0127 | Reserved | | | | | | | | | |
| 0x0128 | PCFG2 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0129 ... 0x013B | Reserved | | | | | | | | | |
| 0x013C | PCFG3 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x013D ... 0x014F | Reserved | | | | | | | | | |
| 0x0150 | PCFG4 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0151 ... 0x0163 | Reserved | | | | | | | | | |
| 0x0164 | PCFG5 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0165 ... 0x0177 | Reserved | | | | | | | | | |
| 0x0178 | PCFG6 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |
| 0x0179 ... 0x018B | Reserved | | | | | | | | | |
| 0x018C | PCFG7 | 7:0 | | | | PTYPE[2:0] | | BK | PTOKEN[1:0] | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.13.1. Host Pipe n Configuration

**Name:** PCFGn
**Offset:** 0x0100 + n*0x14 [n=0..7]
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 34-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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.

### 34.13.2. Bus Access Period of Pipe

**Name:** BINTERVAL - HOST_PIPE Mode
**Offset:** 0x0103
**Reset:** 0x00
**Property:** RW

**Table 34-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 from the type of pipe the desired period is defined as:

| 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.

### 34.13.3. End Point Pipe Status Clear

**Name:** PSTATUSCLR
**Offset:** 0x0104
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 34-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.13.4. End Point Pipe Status Set

**Name:** PSTATUSSET
**Offset:** 0x0105
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 34-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.13.5. End Point Pipe Status

**Name:** PSTATUS - HOST_PIPE Mode
**Offset:** 0x0106
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 34-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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. |

**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.

### 34.13.6. Host Pipe Interrupt Flag register

**Name:** PINTFLAG
**Offset:** 0x0107
**Reset:** 0x00
**Property:** -

**Table 34-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | STALL | TXSTP | PERR | TRFAIL | | TRCPT |
| Access | | | R/W | R/W | R/W | R/W | | R/W |
| Reset | | | 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.

**Bit 0 – 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.

### 34.13.7. Pipe Interrupt Flag Clear

**Name:** PINTENCLR
**Offset:** 0x0108
**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 | | TRCPT |
| Access | | | R/W | R/W | R/W | R/W | | R/W |
| Reset | | | 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.

**Bit 0 – 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.

### 34.13.8. Host Pipe Interrupt Flag Set

**Name:** PINTENSET
**Offset:** 0x0109
**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 | | TRCPT |
| Access | | | R/W | R/W | R/W | R/W | | R/W |
| Reset | | | 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. |

**Bit 0 – 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.

| 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.

## 34.14. Host Registers - Pipe RAM Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | ADDR - Host0 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x04 | PCKSIZE0 | 7:0 | | | | | | | | |
| | | 15:8 | MULTI_PACKET_SIZE[1:0] | | BYTE_COUNT[5:0] | | | | | |
| | | 23:16 | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| | | 31:24 | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| 0x08 | EXTREG - HOST | 7:0 | VARIABLE[3:0] | | | | SUBPID[3:0] | | | |
| | | 15:8 | | VARIABLE[10:4] | | | | | | |
| 0x0A | ADDR - Host1 | 7:0 | ADDR[7:0] | | | | | | | |
| | | 15:8 | ADDR[15:8] | | | | | | | |
| | | 23:16 | ADDR[23:16] | | | | | | | |
| | | 31:24 | ADDR[31:24] | | | | | | | |
| 0x0A | STATUS_BK - Host0 | 7:0 | | | | | | | ERRORFLOW | CRCERR |
| 0x0C | CTRL_PIPE | 7:0 | | PDADDR[6:0] | | | | | | |
| | | 15:8 | PERMAX[3:0] | | | | PEPNUM[3:0] | | | |
| 0x0E | PCKSIZE1 | 7:0 | | | | | | | | |
| | | 15:8 | MULTI_PACKET_SIZE[1:0] | | BYTE_COUNT[5:0] | | | | | |
| | | 23:16 | MULTI_PACKET_SIZE[9:2] | | | | | | | |
| | | 31:24 | AUTO_ZLP | SIZE[2:0] | | | MULTI_PACKET_SIZE[13:10] | | | |
| 0x0E | STATUS_PIPE0 | 7:0 | ERCNT[2:0] | | | CRC16ER | TOUTER | PIDER | DAPIDER | DTGLER |
| | | 15:8 | | | | | | | | |
| 0x12 ... 0x13 | Reserved | | | | | | | | | |
| 0x14 | STATUS_BK - Host1 | 7:0 | | | | | | | ERRORFLOW | CRCERR |
| 0x15 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | STATUS_PIPE1 | 7:0 | ERCNT[2:0] | | | CRC16ER | TOUTER | PIDER | DAPIDER | DTGLER |
| | | 15:8 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.14.1. Host Bank, Address of Data Buffer

**Name:** ADDR - Host
**Offset:** 0x00 + ost*0x04 [ost=0..1]
**Reset:** 0xxxxxxxx
**Property:** NA

**Table 34-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.14.2. Host Bank, Packet Size

**Name:** PCKSIZE
**Offset:** 0x04 + n*0x04 [n=0..1]
**Reset:** 0xxxxxxxx
**Property:** NA

**Table 34-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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[5:0] | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

**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 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]** Pipe size
These bits contains the size of the pipe.
Theses 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) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| **SIZE** (continued) | |
|---|---|
| **SIZE[2:0]** | **Description** |
| 0x7 | 1024 Byte in HS mode(1)<br>1023 Byte in FS mode(1) |

1. For Isochronous pipe only.

**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 should 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 13:8 – BYTE_COUNT[5:0]** 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.

### 34.14.3. Host Bank, Extended

**Name:** EXTREG - HOST
**Offset:** 0x08
**Reset:** 0xxxxxxxx
**Property:** NA

**Table 34-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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
To support the USB2.0 Link Power Management addition the VARIABLE field should be set as described below.

| VARIABLE | Description |
|---|---|
| VARIABLE[3:0] | bLinkState |
| VARIABLE[7:4] | BESL (See LPM ECN) |
| VARIABLE[8] | bRemoteWake |
| VARIABLE[10:9] | Reserved |

**Bits 3:0 – SUBPID[3:0]** SUBPID field send with extended token


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 34.14.4. Host Bank, Status of Bank

**Name:** STATUS_BK - Host
**Offset:** 0x0A + ost*0x0A [ost=0..1]
**Reset:** 0xxxxxxxx
**Property:** NA

**Table 34-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.14.5. Host Bank, Host Control Pipe

**Name:** CTRL_PIPE
**Offset:** 0x00C
**Reset:** 0xXXXX
**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

**Table 34-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

### 34.14.6. Host Bank, Host Status Pipe

**Name:** STATUS_PIPE
**Offset:** 0x0E + n*0x02 [n=0..1]
**Reset:** 0xxxxxxxx
**Property:** PAC Write-Protection, Write-Synchronized, Read-Synchronized

**Table 34-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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.

| Value | Description |
|---|---|
| 0 | No Data Toggle Error. |
| 1 | Data Toggle Error detected. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 35. Configurable Custom Logic (CCL)

## 35.1. Overview

The Configurable Custom Logic (CCL) is a programmable logic peripheral which can be connected to the device pins, events, or 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. This device contains 4LUTs.

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.

Refer to *Configuration Summary* for the number of CCL instances on this device.

## 35.2. Features

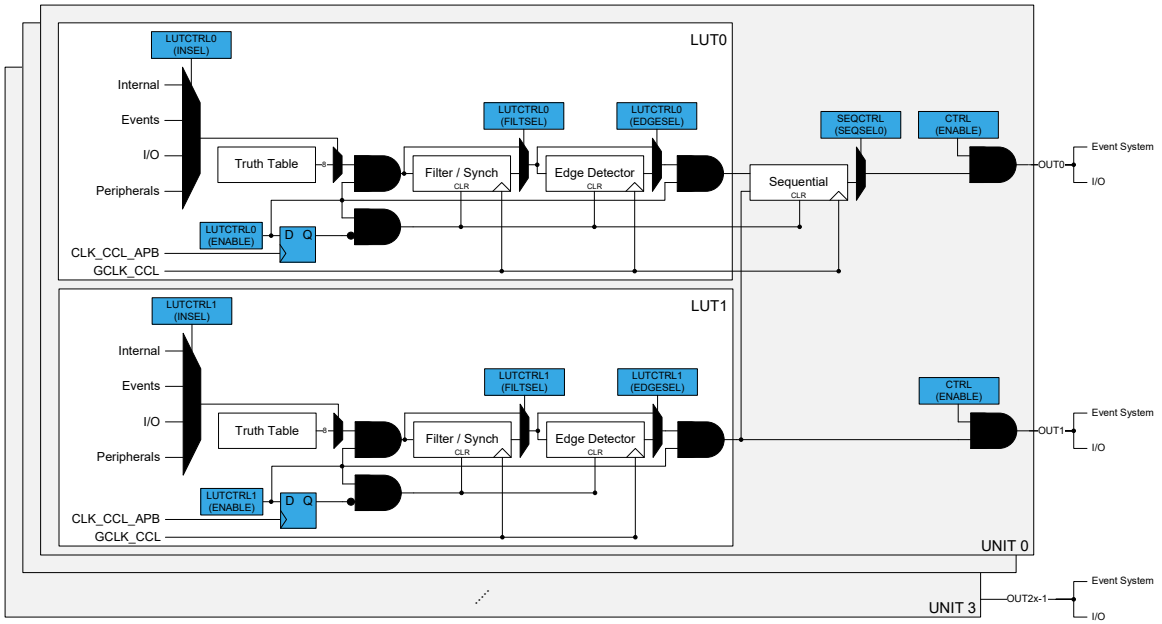
The following are key features of the CCL module:

- Glue logic for general purpose PCB design
- Four 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.

## 35.3. Block Diagram

**Figure 35-1.** Configurable Custom Logic




## 35.4. 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.

## 35.5. 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).

## 35.6. 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.

## 35.7. Clocks

The CCL bus clock (CLK_CCL_APB) can be enabled and disabled in the Main Clock module, MCLK (refer to the Main Clock (MCLK)), 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 the Generic Clock Controller (GCLK) for additional information.

This generic clock is asynchronous to the user interface clock (CLK_CCL_APB).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 35.8. 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 Chapter Event System (EVSYS) for additional information about the event users and event generators.

## 35.9. 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.

## 35.10. Register Access Protection

All registers with write access can be write-protected optionally by the Peripheral Access Controller (PAC). Refer to the Chapter Peripheral Access Controller (PAC) for additional information.

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.

## 35.11. Functional Description

### 35.11.1. 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 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.

### 35.11.2. Operation

#### 35.11.2.1. Initialization

The following bits are enable-protected, meaning that they can only be written when both the CCL and the corresponding even LUT are disabled (CTRL.ENABLE=0 and 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 both the CCL and the corresponding even LUT are disabled (CTRL.ENABLE=0 and 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.

#### 35.11.2.2. 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 Control (CTRL) register for details.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** When the Peripheral Access Controller (PAC) is enabled for the CCL module (WRCTRL.PERID) and a Software Reset (CTRL.SWRST) is executed, a PAC protection error will be generated. If the PAC interrupt is enabled, the interrupt flag for the CCL module (INTFLAGC.CCL) should be cleared.

### 35.11.2.3. 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 the following figure. 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 35-2.** Truth Table Output Value Selection

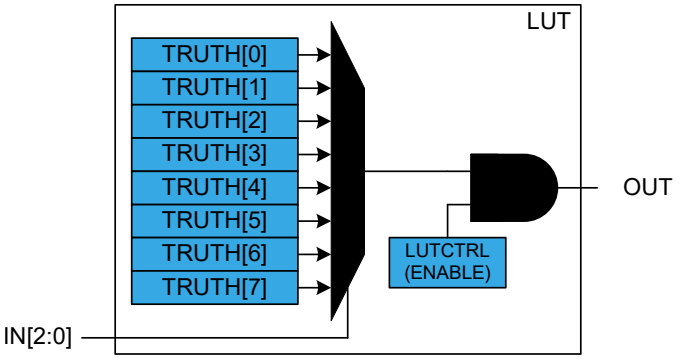



**Table 35-1.** Truth Table of LUT

| IN[2] | IN[1] | IN[0] | OUT |
|---|---|---|---|
| 0 | 0 | 0 | TRUTH[0] |
| 0 | 0 | 1 | TRUTH[1] |
| 0 | 1 | 0 | TRUTH[2] |
| 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] |

### 35.11.2.4. 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

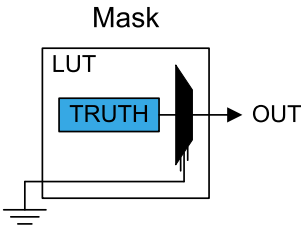
The Input Selection for each input 'y' of LUT x is configured by writing the Input 'y' Source Selection bit in the LUT x Control register (LUTCTRLx.INSELy).

#### 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:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 35-3.** 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:

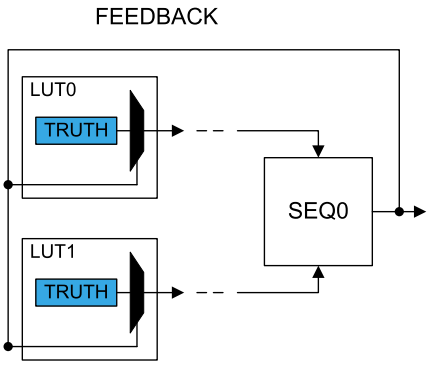
$IN[2N][i] = \mathrm{SEQ}[N]$

$IN[2N+1][i] = \mathrm{SEQ}[N]$

With *N* representing the sequencer number and *i*=0,1,2 representing the LUT input index.

For additional information, refer to Sequential Logic.

**Figure 35-4.** Feedback Input Selection

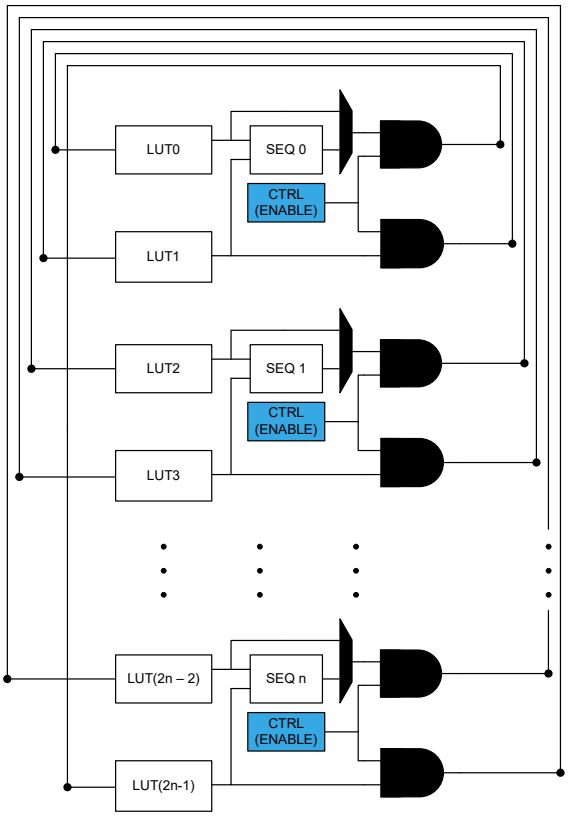



### Linked LUT (LINK)

When selected (LUTCTRLx.INSELy=LINK), the subsequent LUT output is used as the LUT input (for example, LUT2 is the input for LUT1), as shown in the figure below:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

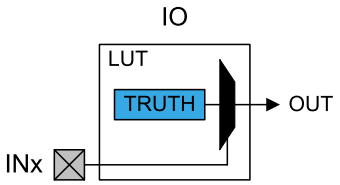
**Figure 35-5.** Linked LUT Input Selection




## I/O Pin Inputs (IO)

When the I/O 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 35-6.** 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]
```

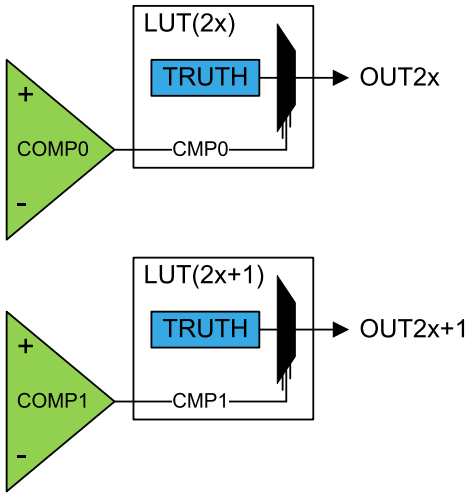
With *N* representing the LUT number and *i*=[0,1,2] representing the LUT input index.

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 35-7.** AC 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 (that is, IN0 is connected to WO0, IN1 to WO1, and IN2 to WO2), as shown in the figure below.

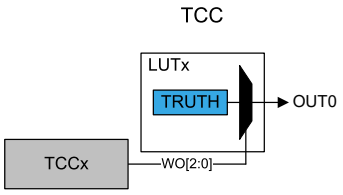
**Note:**
The TCC selection for each LUT follows the formula:

$\mathrm{IN}[N][i] = TCC[N \,\%\, \mathrm{TCC_Instance_Number}].\mathrm{WO}[\mathrm{i}]$

Where *N* represents the LUT number and i represents the LUT input index (i=0,1,2).

Before selecting the waveform outputs, the TCC must be configured first.

**Figure 35-8.** 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:

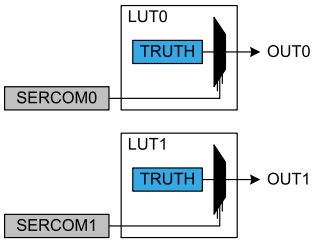
$\mathrm{IN}[N][i] = SERCOM[N \,\%\, \mathrm{SERCOM_Instance_Number}]$

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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 35-9.** SERCOM Input Selection

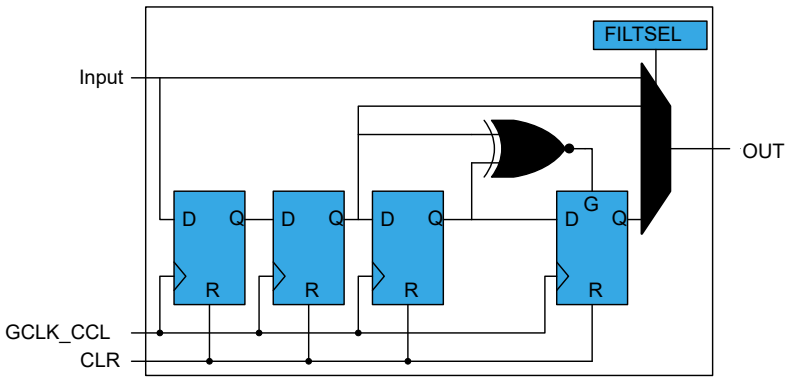



### 35.11.2.5. Filter

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.

**Figure 35-10.** Filter



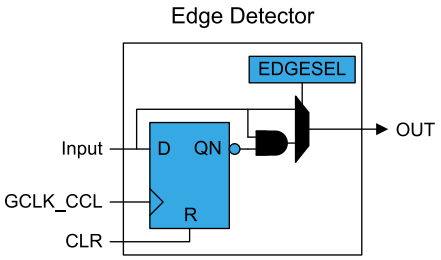

### 35.11.2.6. Edge Detector

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 35-11.** Edge Detector





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.11.2.7. Sequential Logic

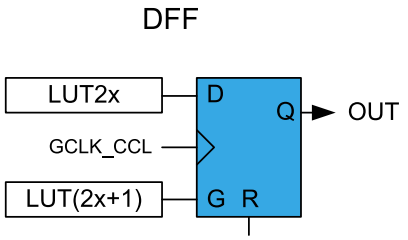
Each LUT pair can be connected to the internal sequential logic which can be configured to work as D flip flop, JK flip flop, gated D-latch or RS-latch by writing the Sequential Selection bits on the corresponding Sequential Control x register (SEQCTRLx.SEQSEL). Before using sequential logic, the GCLK_CCL 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 the following figure.

**Figure 35-12.** 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 the following table.

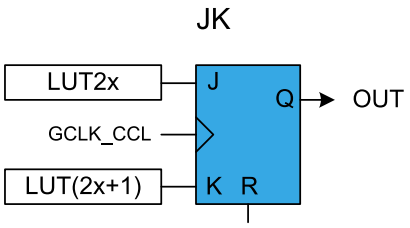
**Table 35-2.** DFF Characteristics

| 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 the following figure.

**Figure 35-13.** 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 the following table.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

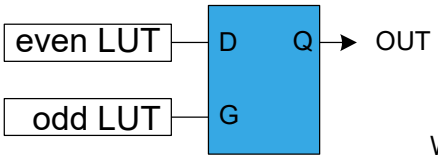
**Table 35-3.** JK Characteristics

| 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 the following figure.

**Figure 35-14.** 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 the following table.

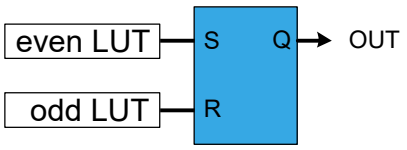
**Table 35-4.** D-Latch Characteristics

| 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 the following figure.

**Figure 35-15.** 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 the following table.

**Table 35-5.** RS-Latch Characteristics

| S | R | OUT |
|---|---|---|
| 0 | 0 | Hold state (no change) |
| 0 | 1 | Clear |
| 1 | 0 | Set |
| 1 | 1 | Forbidden state |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 35.12. Events

The CCL can generate the following output events:

- LUTOUTn with n=0-3: Lookup Table Output Value

Writing a '1' to the LUT Control Event Output Enable bit (LUTCTRLx.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:

- INSELy: The event is used as input for the TRUTH table.

Writing a '1' to the LUT Control Event Input Enable bit (LUTCTRLx.LUTEI) enables the corresponding action on input event. Writing a '0' to this bit disables the corresponding action on input event.

## 35.13. 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.

## 35.14. Peripheral Dependencies

**Table 35-6.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| CCL0 | 0x4482E000 | | **APB:**<br>CLKMSK[1]<br>MASK20 | PCHCTRL[36] | 42<br>STATUS1<br>PERID10 | | Generators:<br>**LUTOUT0:**<br>CHANNEL[83]<br>**LUTOUT1:**<br>CHANNEL[84]<br>**LUTOUT2:**<br>CHANNEL[85]<br>**LUTOUT3:**<br>CHANNEL[86]<br>Users:<br>**LUTIN0:**<br>USER[53],Resynch<br>**LUTIN1:**<br>USER[54],Resynch<br>**LUTIN2:**<br>USER[55],Resynch<br>**LUTIN3:**<br>USER[56],Resynch | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 35-6.** Peripheral Dependencies (continued)

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| CCL1 | 0x44830000 | | **APB:**<br>CLKMSK[1]<br>MASK21 | PCHCTRL[37] | 43<br>STATUS1<br>PERID11 | | Generators:<br>**LUTOUT0:**<br>CHANNEL[87]<br>**LUTOUT1:**<br>CHANNEL[88]<br>**LUTOUT2:**<br>CHANNEL[89]<br>**LUTOUT3:**<br>CHANNEL[90]<br>Users:<br>**LUTIN0:**<br>USER[57],Resynch<br>**LUTIN1:**<br>USER[58],Resynch<br>**LUTIN2:**<br>USER[59],Resynch<br>**LUTIN3:**<br>USER[60],Resynch | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 35.15. 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] | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.15.1. Control

**Name:** CTRL
**Offset:** 0x0
**Reset:** 0x00
**Property:** PAC Write-Protection

**Note:** CTRL register (except the bits ENABLE & SWRST) is Enable Protected when CCL.CTRL.ENABLE = 1.

| 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" section.

**Important:** This bit must be written before enabling the CCL.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Generic clock is not required in standby sleep mode |
| 1 | ENABLE | Generic clock is required in standby sleep mode |

**Bit 1 – ENABLE** Enable

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | The peripheral is disabled |
| 1 | ENABLE | 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 | Name | Description |
|---|---|---|
| 0 | DISABLE | The peripheral is not reset |
| 1 | ENABLE | The peripheral is reset |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.15.2. SEQ Control x

**Name:** SEQCTRL
**Offset:** 0x04 + n*0x01 [n=0..1]
**Reset:** 0x00
**Property:** PAC Write-Protection, Enable-Protected

**Note:** SEQCTRLn register is Enable-protected when CTRL.ENABLE = 1 or LUTCTRLn.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 |
|---|---|---|
| 0 | DISABLE | Sequential logic is disabled |
| 1 | DFF | D flip flop |
| 2 | JK | JK flip flop |
| 3 | LATCH | D latch |
| 4 | RS | RS latch |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 35.15.3. LUT Control x

**Name:** LUTCTRL
**Offset:** 0x08 + n*0x04 [n=0..3]
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-protected

**Note:** LUTCTRLn register is Enable Protected when CCL.LUTCTRLn.ENABLE = 1.

**Table 35-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 Value
These bits define the value of truth logic as a function of inputs IN[2:0].

**Bit 22 – LUTEO** LUT Event Output Enable

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | LUT event output is disabled |
| 1 | ENABLE | LUT event output is enabled |

**Bit 21 – LUTEI** LUT Event Input Enable

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | LUT incoming event is disabled |
| 1 | ENABLE | LUT incoming event is enabled |

**Bit 20 – INVEI** Inverted Event Input Enable

| Value | Name | Description |
|---|---|---|
| 0 | NORMAL | Incoming event is not inverted |
| 1 | INVERTED | Incoming event is inverted |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 8:11, 12:15, 16:19 – INSELx** LUTn Input Source Selection [n=0..2]
These bits select the LUTn input source for CCLm:

**Table 35-8.** CCL Internal Connection

| | 0x4 | 0x5 | 0x6 | 0x7 | 0x8 | 0x9 | 0xA |
|---|---|---|---|---|---|---|---|
| LUT0.IN0 | CCLm_IN0 | AC0 output | SERCOM0 padout[0] | TCC0 WO0 | TCC0 WO4 | TCC4 WO0 | TCC5 WO0 |
| LUT0.IN1 | CCLm_IN1 | AC0 output | SERCOM0 padout[0] | TCC0 WO1 | TCC0 WO5 | TCC4 WO1 | TCC5 WO1 |
| LUT0.IN2 | CCLm_IN2 | AC0 output | SERCOM0 padout[0] | RESERVED | TCC0 WO6 | TCC4 WO0 | TCC5 WO0 |
| LUT1.IN0 | CCLm_IN3 | AC1 output | SERCOM1 padout[0] | TCC1 WO0 | RESERVED | TCC6 WO0 | RESERVED |
| LUT1.IN1 | CCLm_IN4 | AC1 output | SERCOM1 padout[0] | TCC1 WO1 | RESERVED | TCC6 WO1 | RESERVED |
| LUT1.IN2 | CCLm_IN5 | AC1 output | SERCOM1 padout[0] | RESERVED | RESERVED | TCC6 WO0 | RESERVED |
| LUT2.IN0 | CCLm_IN6 | AC0 output | SERCOM2 padout[0] | TCC2 WO0 | RESERVED | TCC0 WO0 | TCC1 WO0 |
| LUT2.IN1 | CCLm_IN7 | AC0 output | SERCOM2 padout[0] | TCC2 WO1 | RESERVED | TCC0 WO1 | TCC1 WO1 |
| LUT2.IN2 | CCLm_IN8 | AC0 output | SERCOM2 padout[0] | RESERVED | RESERVED | TCC0 WO0 | TCC1 WO0 |
| LUT3.IN0 | CCLm_IN9 | AC1 output | SERCOM3 padout[0] | TCC3 WO0 | RESERVED | TCC2 WO0 | TCC3 WO0 |
| LUT3.IN1 | CCLm_IN10 | AC1 output | SERCOM3 padout[0] | TCC3 WO1 | RESERVED | TCC2 WO1 | TCC3 WO1 |
| LUT3.IN2 | CCLm_IN11 | AC1 output | SERCOM3 padout[0] | RESERVED | RESERVED | TCC2 WO0 | TCC3 WO0 |

| Value | Name | Description |
|---|---|---|
| 0x0 | MASK | Masked input |
| 0x1 | FEEDBACK | Feedback input source |
| 0x2 | LINK | Linked LUT input source |
| 0x3 | EVENT | Event input source |
| 0x4 | IO | I/O pin input source |
| 0x5 | AC | AC input source |
| 0x6 | SERCOM | SERCOM input source |
| 0x7 | TCC | TCC input source |
| 0x8 | TCC | TCC input source |
| 0x9 | TCC | TCC input source |
| 0xA | TCC | TCC input source |
| 0xB – 0xF | Reserved | Reserved |

**Bit 7 – EDGESEL** Edge Selection

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Edge detector is disabled |
| 1 | ENABLE | Edge detector is enabled |

**Bits 5:4 – FILTSEL[1:0]** Filter Selection
These bits select the LUT output filter options:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Filter Selection

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Filter disabled |
| 1 | SYNCH | Synchronizer enabled |
| 2 | FILTER | Filter enabled |

**Bit 1 – ENABLE** LUT Enable

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | LUT block is disabled |
| 1 | ENABLE | LUT block is enabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 36. Analog to Digital Converter (ADC)

## 36.1. Overview

The Analog-to-Digital Convertor can convert 1 independent signal using the SAR (Successive Approximation Register) ADC Module. The ADC module supports up to 12 bit resolution. For sample timing and conversion rates, see the *Analog-to-Digital Converter (ADC) Electrical Specifications*. Single-ended and differential measurements are supported. Signed and unsigned sampling formats are supported.

The SAR ADC module can support up to 16 separate analog input channels.

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 AVDD and external VREFH reference voltages can be used (that is, AVDD or VREFH).

The ADC has a digital result compare function for accurate monitoring of user-defined thresholds, with minimum software intervention required. Hardware oversampling is also supported in the ADC's hardware.

**Note:** If AVDD < 2.5V, the user must enable analog charge pumps in SUPC.VREGCTRL.CPEN[1:0].

## 36.2. Features

The Multi-Shared-Channel digital Controller in conjunction with the Microchip Internal Analog ADC SAR core implements these features:

- Up to 12-bit resolution of the numerical output, signed or unsigned
- Signed outputs left-adjusted or right-adjusted
- Up to 4.5 Msps conversion rate per channel
- Up to 16 analog inputs.
- Single-ended and/or differential-ended inputs on a per channel basis for all channels.
- Up to 16 trigger sources, off-chip hardware or on-chip hardware or software generated
- Edge or level active triggering modes, to generate single conversions or bursts of conversions
- A scan trigger per each shared Analog ADC SAR Core to start a scan cycle which can individually include, or not, any of the analog inputs assigned to that Analog ADC SAR Core
- Any channel can select any of the 16 trigger sources or the scan trigger as its own trigger sources or the scan trigger as its own trigger
- The scan trigger itself can select any of the 16 trigger sources as its own source
- Programmable sampling time per Analog ADC SAR Core
- Each Analog Input or Channel Output register can be read from a general dedicated APB output register (user writes the CORDYID and CHRDYID then reads the CHRDYDATA register)
- Optional FIFO-buffered OCP-based DMA Bus Mastering Interface for all channels (Open Core Protocol (OCP))
- APB-accessible FIFO for all channels (intended for systems without the DMA bus mastering interface option). Intended to be used with the system DMA.

**Note:** When using the on-chip triggers (i.e., timers) we suggest that the same GCLK source be used for the ADC controller and Trigger source( i.e., timer). This will insure the ADC will have a constant trigger delay.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 36.3. Block Diagram

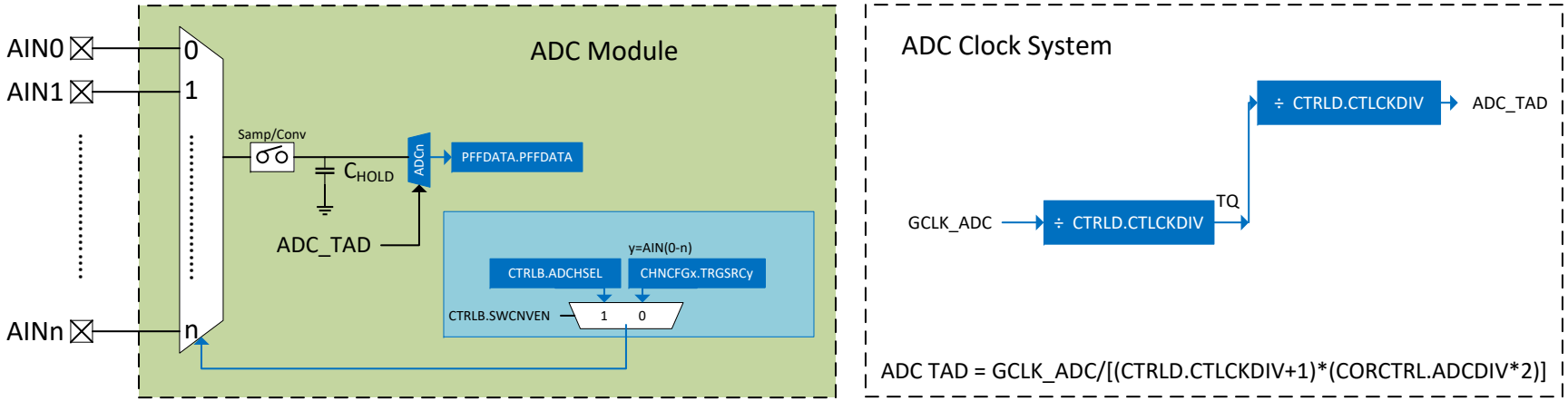
The ADC Module has a dedicated Analog Input Multiplexer, which can select up to 16 different analog inputs . The ADC Module has a Post Processing Block, consisting of a dedicated Digital Filter and a dedicated Digital Comparator.

The ADC's sample data is available in the ADC Channel Ready Data Register (CHRDYDAT) and a 16-sample-deep FIFO (PFFDATA).

The ADC module provides two interrupts to the NVIC: one Global Interrupt and one ADC Module-specific interrupts. Global interrupts are serviced by the CTLINTENCLR, CTLINTENSET, and CTLINTFLAG registers, while the ADC module's interrupts are serviced by INTENCLR, INTENSET, and INTFLAG registers.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 36-1.** ADC Block Diagram

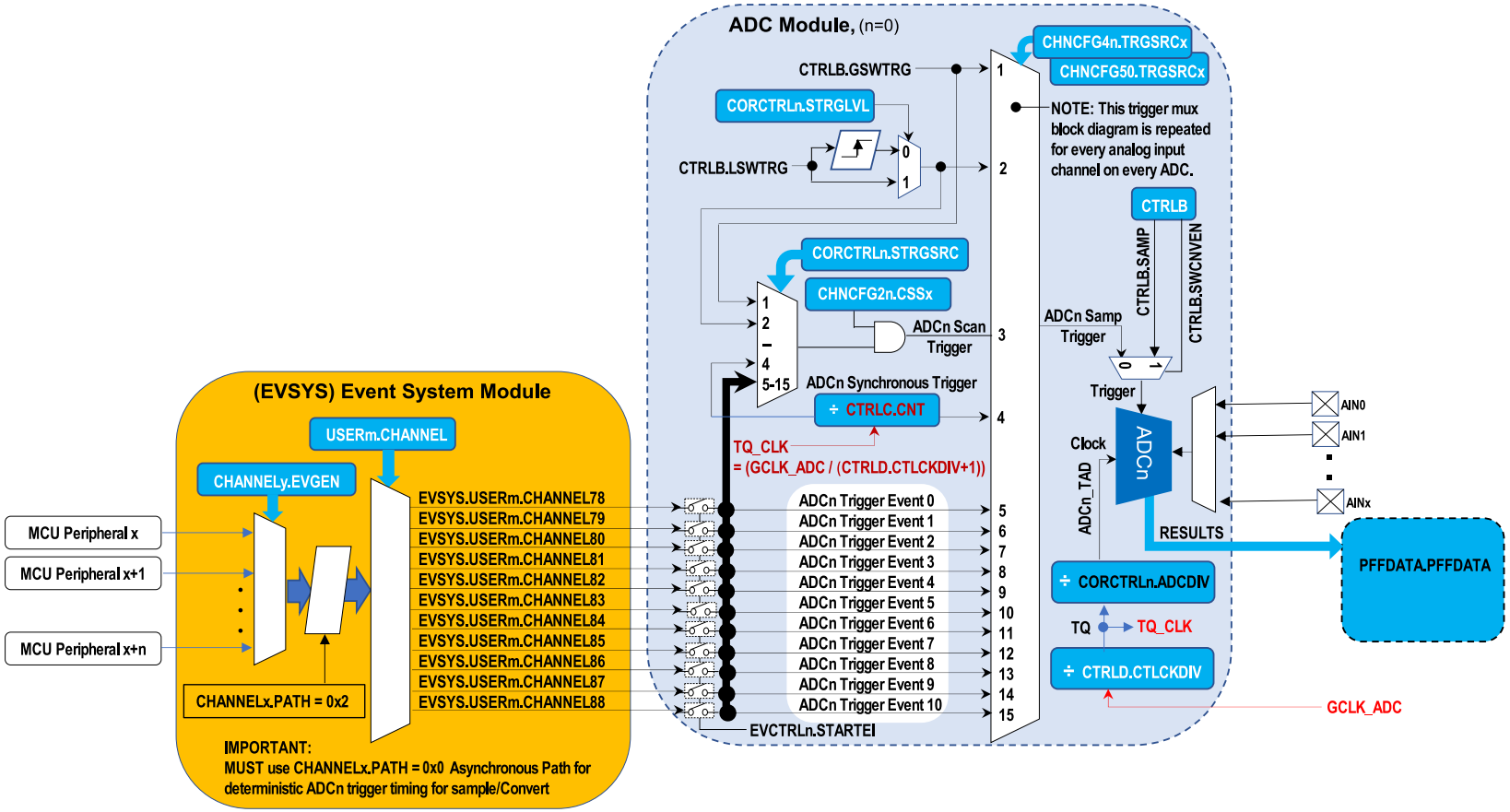



© 2026 Microchip Technology Inc. and its subsidiaries



Downloaded from Arrow.com.

**Figure 36-2.** ADC Trigger Block Diagram




© 2026 Microchip Technology Inc. and its subsidiaries



Downloaded from Arrow.com.

The ADC runs on a peripheral clock provided by the Generic Clock (GCLK) module, known as the GCLK_ADC, and identified as ADC Control Clock, or CTL_CLK in this chapter. The ADC Module has a clock derived from the ADC Control Clock, CORE_CLK. ADC Special Function Registers (SFRs) are identified as to which clock domain they belong: “APB_CLK” for the APB/Main Clock and “GCLK” for the clock derived from the GCLK, i.e. the ADC Control Clock (CTL_CLK).

The ADC Data Bus is 19 bits wide: { ChannelValid, CoreChannelID[5:0], ChannelData{11:0] }. The ChannelValid signal identifies whether the core’s time slot contains valid data. The input channel index is reported in the CoreChannelID bits. The captured signal is reported in the 12 bits of ChannelData.

The ADC Data Bus sends data to the ADC’s status and data registers as well as to the Digital Filter and Digital Comparator Post Processing Blocks. The bus also supports sending filtered results to the ADC’s data registers.

## 36.4. Signal Description

**Table 36-1.** Signal Description

| Signal | Type | Description |
|---|---|---|
| AVDD | Power | Analog Power Supply |
| AVSS | Power | Analog Ground Supply |
| ADC_VREFH | Reference | External voltage reference |
| ADC AINx, x =0,1,,...,11 | Analog | AINx Analog inputs for ADC Module 0 |

## 36.5. Functional Description

### 36.5.1. Principle of Operation

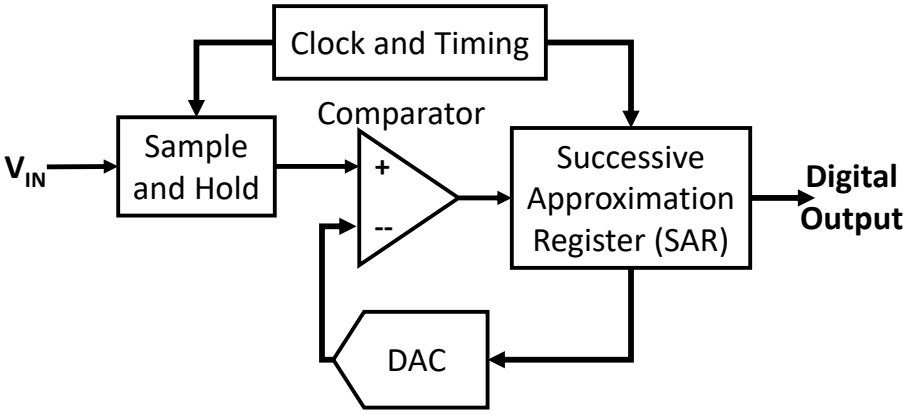
**Important:** When AVDD < 2.5V, it is recommended that the user enable the analog charge pumps when in active mode by setting the SUPC.VREGCTRL.CPEN[1:0] bits. When the CPEN[N] bits are set, the enable and auto-enable of the associated charge pump[N] are set.

The basic architecture of an ADC Module is shown in the following figure.

At the start of a capture cycle the Sample and Hold capacitor is connected to the incoming voltage on a trigger event until the expiration of the sample time defined by CORCTRLx.SAMC. At the end of the sampling period the Sample and Hold is disconnected from the input signal and connected to the Comparator and the conversion sequence begins. ADC conversion time = (# bits +1)*TAD).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 36-3.** ADC Module Architecture




## 36.5.2. Basic Operation

### 36.5.2.1. Enabling, Disabling, and Resetting

The ADC is enabled by setting **CTRLA.ENABLE** = 1. Setting this bit to zero disables the ADC. The ADC can be reset by setting **CTRLA.SWRST** = 1 to initiate a software reset. (The ADC module is reset when the **SYNCYBUSY.SWRST** bit goes low.)

The steps to enable the ADC are defined in the “Initialization” section.

The analog bias in each of the ADC SAR Cores can be powered up / down individually by setting / resetting the corresponding bit CTRLD.ANLENx.

Each ADC SAR Core x has a built-in warm-up counter, which will not allow any trigger to propagate for a warm-up period (typically 20us) . This period occurs when the core is enabled or re-enabled.

**Note:** The analog circuitry must be turned on before the digital circuitry.

The individual CHNENx bits can be used to provide a finer control of the power consumption of the entire ADC module, but they do not control the enabling of the analog and bias circuitry in the ADC Analog IP.

A device reset forces all registers to their reset state. This forces the A/D module to be turned off, and any conversion or sampling sequence is aborted. The bandgap reference circuit is also turned off, so the ADC resuming operations will have to wait for the bandgap reference circuit to stabilize by polling (or requesting to be interrupted by) CTLINTFLAG.VREFRDY status bit.

If the ADC is stopped during debug mode, it is handled by simply resetting the ADC module state machines. This is accomplished by forcing all the internal channel enable signals CTRLD.CHNENx to zero. The CTRLA.ENABLE should stay high, which will enable all the core_clk[x], x=0...(ACN-1), to remain active and keep all the analog biasing circuits alive.

**Note:** The bit **DBGCTRL.DBGRUN** is unchanged by a software reset.

### 36.5.2.2. Operation

A trigger event initiates the start of the ADC sampling of the selected analog input. After the expiration of the SAMC sample time the conversion sequence is automatically initiated by the hardware assuming CTRLB.SWCNVEN = 0. For maximum efficiency, in the case of SCAN mode, at the conclusion of sampling for the current active analog input and while the conversion cycle is in progress, the analog channel input multiplexer will switch to the next analog input in the users scan sequence in preparation to initiate the next channel sampling sequence immediately once the previous channel conversion is complete. In SCAN mode, sample/conversion progression is always lower to higher numbered channels as selected by the user in the CHNCFG2.CSSx register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 36.5.2.3. Clocks Setup

The interface to the Advanced Peripheral Bus (APB) is clocked by the system's Main Clock.

The ADC Core can change the provided GCLK to obtain a maximum frequency ≤72Mhz.
The Peripheral Channel Control Register PCHCTRLn selects the GCLK generator. The register PCHCTRL[n].GEN[m] is configured to select the GCLK's clock generator to be used for the ADC. The resulting ADC Control Clock is referred to as CTL_CLK in this chapter.

See *GCLK Peripheral Channel Control Mapping* and *Peripherals Configuration Summary* for a full peripheral index listing. The generator listing can be found in the *Peripheral Dependencies* table.

This register selects the input clock for the generator and provides the clock divider to divide an input clock down to the clock output of GCLK Generator m.

#### 36.5.2.4. Conversion Timing and Sampling Rate

If the period of CTL_CLK is $T_Q$ , then the period of the respective ADC_CLK module clock, $T_{AD}$ , is given by:

$T_{AD} = 2 \cdot ADCDIV \cdot T_Q$ .

The minimum ADC clock period is $T_{AD} = 2 \cdot T_Q$ .

**Provided CTRLB.SWCNVEN=0:**

The maximum ADC throughput rate for the ADC module with $N_{BITS}$ of resolution, is:

Maximum Throughput rate $F_{TPR}$ = [ 1/ [(CORCTRL.SAMC+2)TAD + ($N_{BITS}$+1)TAD] ] / #Active ADC Channels

**Example:**

(CORCTRL.SAMC=0x1, 12bit resolution and 2 AINx scan channels selected)

$F_{TPR}$ = [ 1/ (3TAD + 13TAD) ] / 2 = (1 / 16TAD) / 2

**Important:** The ADC Clock (CTL_CLK) must have the following characteristics:

1. $F_{AD} \leq 72MHz$

#### 36.5.2.5. Voltage Reference Selection

The ADC's voltage reference is selected by CTRLD.VREFSEL[2:0]:

| VREFSEL[2:0] | ADREF+ | ADREF- |
|---|---|---|
| 000 | $A_{VDD}$ | $A_{VSS}$ |
| 001 | External $V_{REF}H$ | $A_{VSS}$ |
| 010 – 111 | Reserved | Reserved |

#### 36.5.2.6. ADC Resolution

The output resolution is selected by CORCTRL.SELRES, to choose between 6 bits, 8 bits, 10 bits, or 12 bits (default). Changing the resolution of the ADC module will also change the ADC throughput rate.

**Note:** Required analog signal sample time in ADC clock cycles (TAD) as defined in CORCTRL.SAMC depend on source impedance of external signal source. Consult electrical specifications for required sample time based on external source impedance.

#### 36.5.2.7. Differential and Single Ended Conversions

The ADC can measure up to 12 single ended input channels (AIN[(n-1):0]; n=12), or up to 3 differential inputs channels

Refer to the CHNCFG3 register for bit descriptions on how to configure them for differential mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.5.2.8. Input Channel Conversion Priority

The selection of the next analog input for conversion uses a “natural ascending order” priority scheme to resolve simultaneous requests for input conversions. In scan mode, analog input channels are always processed from lower to higher channels as selected by the user in the ADC CHNCFG2n.CSS register.

### 36.5.2.9. Output Data Formats

The Analog/Digital core provides up to 12-bits of resolution which are available in registers CHRDYDAT and PFFDATA as 16 bits. The 12-bit ADC output data is translated into a 16-bit format specified by CHNCFG2.FRACTk. Additionally, the CHNCFG2.SIGNk bit specifies the data format as unsigned, or two’s complementary signed.

The following table shows the four cases of the channel index k 16-bit output data format depending on the values [ FRACTk ,SIGNk ]:

**Table 36-2.** ADC Analog Input Channel n Data Format Examples

| FRACT*k* | SIGN*k* | 16-bit Output Data Format |
|---|---|---|
| - | - | s = sign bit<br>d = mantissa value bit |
| 0 | 0 | 0000 dddd dddd dddd |
| 0 | 1 | ssss sddd dddd dddd |
| 1 | 0 | dddd dddd dddd 0000 |
| 1 | 1 | sddd dddd dddd 0000 |

### 36.5.2.10. Initialization

The following steps are required to configure the ADC for operation:

1. Configure input pins for operation as analog input pins, ADC_AINk.
2. Configure a GCLK generator to provide the GCLK_ADC clock .
3. Configure ADC registers:
   a. Copy factory-provided calibration values from the CAL OTP Register FCCFG65 into CALCTRL for all the ADC in use by the application.
   b. Configure CTRLA as needed but do not set CTRLA.ENABLE.
   c. Configure CTRLD, choosing the clock divider (CTLCKDIV) for the CTL_CLK clock based on the GCLK_ADC, Voltage Reference (VREFSEL), and Wake-up Delay Exponent (WKUPEXP).
   d. (Optionally) Enable ADC interrupts for FIFO support using CTLINTENSET. Optionally enable the interrupts for VREFRDY and Core n ready. (Or these conditions can be monitored by watching CTLINTFLAG in a while(1) loop.) Enable interrupt vector (ADC Global Interrupt) in the NVIC.
   e. Enable ADC interrupts using the INTENSET registers. Enable the corresponding NVIC interrupt vector.
   f. For each ADCn, configure its Control register CORCTRLn:
      - Set the clock divider, ADCDIV, for the ADC Module clock, CORE_CLK[n].
      - Set the sample count SAMC (ADCDIV and SAMC determine the ADC Module’s sample rate fs).
      - If ADC resolution other than 12 bits is desired, set it using SELRES.
      - Other settings in CORCTRLn as needed.
   g. Setup differential channels using CHNCFG3n.DIFFk = 1 for signals pairs k and k+1 (k even).
   h. Select the input channels for each scan using CHNCFG2n.CSSk = 1.
   i. Set the data format for each sample using CHNCFG2n.FRACTk and CHNCFG3n.SIGNk.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

j. Setup channel triggers.
   - (Optional: Setup up the Digital Filter and Digital Comparator with the associated modules).

4. Enable the ADC by setting CTRLA.ENABLE = 1.
5. Wait until CTLINTFLAG.VREFRDY = 1, either using an ISR or polling in a while(1) loop. This signals that the chip's analog environment is ready.
6. Wait for CRRDYn to go high in CTLINTFLAG, indicating that the corresponding ADC Module is ready. This can be polled in a while(1) loop or trapped by the ISR attached to the ADC Global Interrupt.

#### 36.5.2.11. Channel Triggers

The ADC controller starts the capture and conversion of an input channel when that channel is "triggered". The ADC Controller needs to know when a channel requests an analog-to-digital conversion. For this purpose, each input channel contains a trigger signal which is passed by the channel to the Module's Access Arbiter. The Access Arbiter drives the address selections to both the input Analog Multiplexer and to the output Digital De-multiplexer.

##### 36.5.2.11.1. Trigger Priority

When two triggers arrive at the Access Arbiter at the same time the trigger associated with the input channel having the lower index *k* will be serviced. The second trigger will be queued for later service, provided it is not overwritten by a third trigger request. In that case, the third trigger will be lost and not serviced.

##### 36.5.2.11.2. Channel Trigger Selection

The trigger used for the ADC, Input Channel k, is specified by CHNCFG{4|5}n.TRGSRCk[3:0]. (CHNCFG4n contains the trigger choices for k = 0,1,...,7. CHNCFG5n contains the trigger choices for k = 8,9,...,15. TRGSRCk for Sn ≤ k are not defined.)

The possible values for TRGSRCk are:

= 0000: No Trigger (NOP)

= 0001: Global Software Trigger (CTRLB.GSWTRG)

= 0010: Global Level Software Trigger (CTRLB.LSWTRG)

= 0011: SCANTRG - Scan Trigger

= 0100: STRIG Synchronous Trigger

= 0101 - 1111: ADC Trigger Event User 0 – 10

Input channels with no triggers will not be serviced during the Module's operation. For all trigger sources except the SCANTRG, setting the TRGSRCk will result in only input channel k being serviced by the Module. To collect a "scan" of input channels TRGSRCk must be set to SCANTRG and then SCANTRG is defined by CORCTRLn.STRGSRC. All the input channels to be included in a scan started by SCANTRG must have CHNCFG2n.CSSk set to one.

The SCANTRG source is selected by CORCTRLn.STRGSRC:

= 0000: No trigger (NOP)

= 0001: Global Software trigger (CTRLB.GSWTRG)

= 0010: Level Software trigger (CTRLB.LSWTRG)

= 0011: Reserved

= 0100: Synchronous Trigger (STRIG)

= 0101 - 1111: EVSYS Event Generator 0 – 10


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The Synchronous Trigger (STRIG) is driven by a counter at the ADC Control Clock (CTL_CLK) frequency and fires when the counter reaches the value defined by CTRLC.CNT[7:0]. To enable this clock set CTRLB.STRGEN to one.

#### 36.5.2.11.3. Software Triggers

ADC captures can be directly controlled from software by using the Global Software Trigger to start a single capture (when CTRLB.GSWTRG = 1) or by using the Global Level Software Trigger to start a burst of captures that will continue as long CTRLB.LSWTRG = 1 and stop when LSWTRG = 0.

#### 36.5.2.11.4. ADC Debugging

Setting CTRLB.SWCNVEN = 1 allows two bits in CTRLB to control the Module and the input channel specified by CTRLB.ADCHSEL. For this to work all the other input channels for the Module must be disabled by setting TRGSRCk = 0, which can only be accomplished when the ADC is disabled. Sampling of the specified input channel starts when CTRLB.SAMP is set to one and stops when CTRLB.SAMP is reset to zero. Immediately after this start conversion by setting CTRLB.RQCNVRT to one.

The CTRLB register has a dedicated SYNCBUSY bit to allow the manipulation of these bits when the ADC is enabled. After writing to the CTRLB register the SYNCBUSY.CTRLB will go high. Wait until the bit goes low before writing to CTRLB again.

#### 36.5.2.11.5. Trigger Limitations

Trigger Rule 1: In order to ensure synchronizing every single pulse on GSWTRG, the GSWTRG pulses must be at least 4 ADC Control Clock periods apart, positive edge to positive edge (because the GSWTRG lasts only 1 single APB clock period by construction).

Trigger Rule 2: If a channel k is effectively included in scan n by setting CHNCFG{4|5}n.TRGSRCk = 3 (SCANTRG) and CHNCFG2n.CSSk= 1, then the user MUST ensure that NO other triggers are generated for that channel using CTRLB.RQCNVRT or any digital filter. Otherwise the scan behavior is unpredictable.

Trigger Rule 3: In order to ensure synchronizing every single pulse on LSWTRG, the LSWTRG pulses must be at least 8 ADC Control Clock periods + 10 Main Clock clock periods positive edge to positive edge, and also at least 4 ADC control clock periods + 4 Main Clock periods negative edge to positive edge.

### 36.5.3. ADC Result Registers

Each of the 16 analog input channels has an associated results register. To read the converted data write the ADC Module index n and the channel index k in the CORCHDATAID register:

CORCHDATAID.CORDYID = n

CORCHDATAID.CHRDYID = k

Then read the data from the CHRDYDAT register. The captured data bits are stored in CHRDYDAT, together with the FRACT, SIGN, DIFF, and LVL settings associated with this input channel. CHRDYDAT is 16 bits wide. For more information on the data formats found in this field consult *Output Data Formats*.

When data from ADCn, input channel k, is ready to be read by the application the INTFLAGn.CHRDY[k] bit will be set. If the corresponding bit in INTENSETn has been set, then the interrupt associated with ADCn will fire. (To reset the CHRDY bit write one to it.) If the application does not read the new data before another sample arrives the Channel Overwritten Error flag (INTFLAGn.CHNERRC) bit will be set.

### 36.5.4. ADC Result FIFO

The ADC has a 16-sample deep FIFO. To capture ADCn results in this FIFO, first enable the FIFO: PFFCTRL.PFFEN = 1. Then enable the FIFO PFFCTRL.PFFCR0= 1. FIFO results are available in the PFFDATA register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The status of the FIFO is reflected in the CTLINTFLAG register. Setting the corresponding bits in the CTLINTENSET register enables these bits to fire the ADC Global Interrupt.

### 36.5.5. DMA Operation

**Table 36-3.** DMA Event Trigger Mapping

| Instance | Channel | CHCTRLB.TRIGSRC [Index] | Presentation |
|---|---|---|---|
| ADC | PFFRDY | 37 | ADC DMA PFFRDY Trigger |

The DMA can be programmed to read the APB Bus FIFO when there is available data. The register value **PFFCTRL.PFFRDYDMA** controls whether the DMA will be read when the FIFO is half full (0), or Not Empty (1). CTLINTFLAG.PFFHFUL and CTLINTFLAG.PFFRDY will be set when data is to be read.

For the FIFO to work, first enable the FIFO by setting **PFFCTRL.PFFEN** = 1. Next enable the ADC Core by setting **PFFCTRL.PFFCRk** = 1. See the section “ADC Result FIFO” for more information.

### 36.5.6. Interrupts

The ADC module provides two interrupts to the NVIC: one Global Interrupt (REQ0) and one Module-specific interrupts (REQ1). Global interrupts are serviced by the **CTLINTENCLR, CTLINTENSET,** and **CTLINTFLAG** registers, while each ADC Module’s interrupts are serviced by **INTENCLR**, **INTENSET**, and **INTFLAG** registers.

When the ADC’s chosen voltage reference status changes (not ready to ready or the reverse) the **CTLINTFLAG.VREFUPD** bit will be set. For this bit to fire the global interrupt **CTLINTENSET**.**VREFUPD** should be set to one. The ADC’s global interrupt ISR should always check this flag and disable ADC operation when **CTLINTFLAG.VREFUPD** = 1 and **CTLINTFLAG.VREFRDY** = 0, indicating that the voltage reference is no longer “ready” and therefore no longer useable.

When the startup delay for each ADC Module has elapsed a **CRRDY*n*** bit in the **CTLINTFLAG** register will go high. This can trigger a global interrupt as well if the corresponding bit in **CTLINTENSET** has been set. Note that the **CRRDY*n*** bit in INTFLAG will stay high and therefore continue to fire the global interrupt unless the global interrupt service routine disables the Core Ready Interrupts from firing by setting the **CTLINTENCLR.CRRDY*n*** bits to one before exiting.

Other bits in the **CTLINTFLAG** register are related to the operation of the APB Data FIFO. The **CTLINTFLAG.PFFHFUL** bit goes high when the FIFO is half full. The **CTLINTFLAG.PFFRDY** goes high when the FIFO is “ready”, i.e. when it contains data to be ready. FIFO overflow and underflow will set the **CTLINTFLAG.PFFOVF** or **CTLINTFLAG.PFFUNF** bits, respectively.

### 36.5.7. Events

The ADC can generate events, which can be shared by the Event System (EVSYS) module with other modules, and it can be an event user of events generated by other modules and distributed by the EVSYS module. The ADC can generate an event when a Digital Comparator “Hit” occurs. It can also generate an event when a new data result is “ready” (available for reading.) As a user the ADC often uses events generated by a timer to trigger the capture and conversion of new samples.

There is one Results Ready (ADC_RESRDY) generators. Similarly, there is one Comparator Hit (ADC_CMP) generators. The ADC has 16 “trigger” users in the EVSYS (TRIG*(n-1)*, *n* = 16). This provides 16 separate event users which can be used to trigger up to eleven different types of ADC captures.

The ADC Module has a dedicated “ADC Event Control” register (EVCTRL). The register controls the Compare Hit and Results Ready generator. To use generated events as a capture trigger **EVCTRL*n*.STARTEI** must be set. The polarity of trigger inputs can be inverted by setting **EVCTRL*n*. STARTINV.**


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.5.8. Power Management

The ADC will continue to operate in any sleep mode where the selected source clock is running. The ADC'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.

**Note:** If several events are connected to the ADC, the enabled action will be taken on any of the incoming events.

### 36.5.9. Idle and Standby Sleep Modes Operation

The host device supports several modes of operation, starting at Active and ending with Off:

| Mode of Operation: | Behavior: | Wake-Up Signal: |
|---|---|---|
| Active | CPU running.<br>All memory is active.<br>ADC can run. | None |
| (CPU) Idle | CPU halted.<br>All memory is active.<br>ADC (1) | All Interrupts |
| Standby | All logic content retained.<br>ADC (1)<br>NVM in sleep mode.<br>Configurable RAM retention support. | All Interrupts |
| Hibernate | Only backup domain remains active.<br>ADC cannot run.<br>Configurable RAM retention.<br>Only 32 kHz clock can run. | Backup Interrupts |
| Backup | Only backup domain remains active.<br>ADC cannot run.<br>Only 32 kHz clock can run. | Backup Interrupts |
| Off | Everything is off. | External Reset |

**Note:**

1. The CTRLA.ONDEMAND and CTRLA.RUNSTDBY bits control the behavior of the ADC during Idle and Standby sleep modes, in cases where the ADC is enabled (CTRLA.ENABLE = 1). See the following table.

**Table 36-4.** ADC Idle and Standby Mode Behavior

| CTRLA.ENABLE | CTRLA.RUNSTDBY | CTRLA.ONDEMAND | Description |
|---|---|---|---|
| 0 | x | x | Disabled. |
| 1 | 0 | 0 | Run in all sleep modes except STANDBY. |
| 1 | 0 | 1 | Run in all sleep modes on request except STANDBY. |
| 1 | 1 | 0 | Run in all sleep modes. |
| 1 | 1 | 1 | Run in all sleep modes on request. |

**Notes:**

- With On Demand Control, when CTRLA.ONDEMAND = 1, the analog block is always powered-off. When a start request is detected, the system returns from sleep and starts a new conversion.
- With Run in idle/sleep mode, RUNSTDBY is enabled and the GCLK to the ADC continues to run when ONDEMAND = 0.
- With Run in Idle or Sleep mode and both ONDEMAND and RUNSTDBY set, the GCLK will be activated when requested. Therefore, there will be an additional delay until the analog front end is ready. Flag CTLINTFLAG.CRRDY = 1 will indicate when the corresponding ADC is ready.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.5.10. Debug Operation

Setting **DBGCTRL.DBGRUN** = 1 enables the ADC to continue operation when the device is halted in debug mode. If this bit is not set, then the ADC is halted in debug mode.

If **DBGCTRL.DBGRUN**=0, then during debug the ADC macro will capture all the triggers that fire and on exit from the debug operation to normal running operation the ADC will perform conversions on the captured event or events based on ADC channel priorities.

During a debug stop all the ADC registers are still write protected as long as **CTRLA.ENABLE**=1.

### 36.5.11. Register Synchronization

The **SYNCBUSY** register supports the **CTRLA** and **CTRLB** registers. When **CTRLA.ENABLE** has been set no additional writes to this bit are allowed as long as the SYNCYBUSY.ENABLE bit remains high. Additional writes to the **CTRLB** register are not allowed as long as the **SYNCBUSY.CTRLB** bit remains high.

Register synchronization is required for other registers in the GLCK clock domain. These registers can be modified while CTRLA.ENABLE=0, i.e. as long as the ADC is not enabled. Once the ADC is enabled (CTRLA.ENABLE=1) these registers are synchronized to the APB_CLK (Main Clock) domain and are write protected.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 36.6. Peripheral Dependencies

**Table 36-5.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| ADC | 0x44818000 | 130 : REQ0<br>131 : REQ1 | **APB:**<br>CLKMSK[1]<br>MASK17 | PCHCTRL[33] | 39<br>STATUS1<br>PERID7 | 37 : PFFRDY | Generators:<br>**CHRDY:**<br>CHANNEL[76]<br>**CMP:**<br>CHANNEL[77]<br>Users:<br>**TRIG0:**<br>USER[38],Async<br>**TRIG1:**<br>USER[39],Async<br>**TRIG2:**<br>USER[40],Async<br>**TRIG3:**<br>USER[41],Async<br>**TRIG4:**<br>USER[42],Async<br>**TRIG5:**<br>USER[43],Async<br>**TRIG6:**<br>USER[44],Async<br>**TRIG7:**<br>USER[45],Async<br>**TRIG8:**<br>USER[46],Async<br>**TRIG9:**<br>USER[47],Async<br>**TRIG10:**<br>USER[48],Async | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 36.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | ONDEMAND | RUNSTDBY | | AIPMPEN | | ANAEN | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x04 | CTRLB | 7:0 | SAMP | RQCNVRT | | | ADCHSEL[3:0] | | | |
| | | 15:8 | SWCNVEN | | | | STRGEN | TRGSUSP | LSWTRG | GSWTRG |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | CNT[7:0] | | | | | | | |
| | | 15:8 | CNT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0C ... 0x0F | Reserved | | | | | | | | | |
| 0x10 | CTRLD | 7:0 | | | | | | | | |
| | | 15:8 | | | CTLCKDIV[5:0] | | | | | |
| | | 23:16 | | | | ANLEN0 | | | | CHNEN0 |
| | | 31:24 | | VREFSEL[2:0] | | | WKUPEXP[3:0] | | | |
| 0x14 ... 0x1F | Reserved | | | | | | | | | |
| 0x20 | CORCTRL | 7:0 | SAMC[7:0] | | | | | | | |
| | | 15:8 | | EIS[2:0] | | | SELRES[1:0] | | SAMC[9:8] | |
| | | 23:16 | | SCNRTDS | STRGLVL | | STRGSRC[3:0] | | | |
| | | 31:24 | | ADCDIV[6:0] | | | | | | |
| 0x24 | CHNCFG1 | 7:0 | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN |
| | | 15:8 | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN |
| | | 23:16 | LVL | LVL | LVL | LVL | LVL | LVL | LVL | LVL |
| | | 31:24 | LVL | LVL | LVL | LVL | LVL | LVL | LVL | LVL |
| 0x28 | CHNCFG2 | 7:0 | CSS | CSS | CSS | CSS | CSS | CSS | CSS | CSS |
| | | 15:8 | CSS | CSS | CSS | CSS | CSS | CSS | CSS | CSS |
| | | 23:16 | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT |
| | | 31:24 | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT |
| 0x2C | CHNCFG3 | 7:0 | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF |
| | | 15:8 | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF |
| | | 23:16 | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN |
| | | 31:24 | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN |
| 0x30 | CHNCFG4 | 7:0 | TRGSRC1[3:0] | | | | TRGSRC0[3:0] | | | |
| | | 15:8 | TRGSRC3[3:0] | | | | TRGSRC2[3:0] | | | |
| | | 23:16 | TRGSRC5[3:0] | | | | TRGSRC4[3:0] | | | |
| | | 31:24 | TRGSRC7[3:0] | | | | TRGSRC6[3:0] | | | |
| 0x34 | CHNCFG5 | 7:0 | TRGSRC9[3:0] | | | | TRGSRC8[3:0] | | | |
| | | 15:8 | TRGSRC11[3:0] | | | | TRGSRC10[3:0] | | | |
| | | 23:16 | TRGSRC13[3:0] | | | | TRGSRC12[3:0] | | | |
| | | 31:24 | TRGSRC15[3:0] | | | | TRGSRC14[3:0] | | | |
| 0x38 | CALCTRL | 7:0 | CALBITS[7:0] | | | | | | | |
| | | 15:8 | CALBITS[15:8] | | | | | | | |
| | | 23:16 | CALBITS[23:16] | | | | | | | |
| | | 31:24 | CALBITS[31:24] | | | | | | | |
| 0x3C | EVCTRL | 7:0 | | | CMPEO | RESRDYEO | STARTINV | | | STARTEI |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x40 ... 0xAF | 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0xB0 | CMPCTRL | 7:0 | ADCMPLO[7:0] | | | | | | | |
| | | 15:8 | IEBTWN | IELOHI | IELOLO | CMPEN | ADCMPLO[11:8] | | | |
| | | 23:16 | ADCMPHI[7:0] | | | | | | | |
| | | 31:24 | | | IEHIHI | IEHILO | ADCMPHI[11:8] | | | |
| 0xB4 ... 0xBF | Reserved | | | | | | | | | |
| 0xC0 | FLTCTRL | 7:0 | | | | DATA16EN | FMODE | OVRSAM[2:0] | | |
| | | 15:8 | | | FLTCHNID[3:0] | | | | | FLTEN |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xC4 ... 0xCF | Reserved | | | | | | | | | |
| 0xD0 | CORCHDATAID | 7:0 | | | CORDYID[1:0] | | CHRDYID[3:0] | | | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xD4 | CHRDYDAT | 7:0 | CHRDYDAT[7:0] | | | | | | | |
| | | 15:8 | CHRDYDAT[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | FRACT | SIGN | DIFF | LVL |
| 0xD8 | PFFDATA | 7:0 | PFFDATA[7:0] | | | | | | | |
| | | 15:8 | PFFDATA[15:8] | | | | | | | |
| | | 23:16 | PFFFRACT | PFFSIGN | PFFCORID[1:0] | | PFFCHNID[3:0] | | | |
| | | 31:24 | PFFCNT[7:0] | | | | | | | |
| 0xDC ... 0xE3 | Reserved | | | | | | | | | |
| 0xE4 | PFFCTRL | 7:0 | | | | PFFCR0 | | | PFFEN | |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | PFFRDYDM |
| | | 31:24 | | | | | | | | |
| 0xE8 | SYNCBUSY | 7:0 | | | | | | CTRLB | ENABLE | SWRST |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0xEC ... 0xFB | Reserved | | | | | | | | | |
| 0xFC | CTLINTENSET | 7:0 | VREFRDY | VREFUPD | | | | | | CRRDY0 |
| | | 15:8 | | | | | PFFHFUL | PFFRDY | PFFOVF | PFFUNF |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0100 | CTLINTENCLR | 7:0 | VREFRDY | VREFUPD | | | | | | CRRDY0 |
| | | 15:8 | | | | | PFFHFUL | PFFRDY | PFFOVF | PFFUNF |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0104 | CTLINTFLAG | 7:0 | VREFRDY | VREFUPD | | | | | | CRRDY0 |
| | | 15:8 | | | | | PFFHFUL | PFFRDY | PFFOVF | PFFUNF |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x0108 ... 0x011F | Reserved | | | | | | | | | |
| 0x0120 | INTENCLR | 7:0 | SOVFL | | | CMPHIT | | | | |
| | | 15:8 | | | | | EOSRDY | CHNERRC | FLTRDY | CHRDYC |
| | | 23:16 | CHRDY[7:0] | | | | | | | |
| | | 31:24 | CHRDY[15:8] | | | | | | | |


© 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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x0124 | INTENSET | 7:0 | SOVFL | | | CMPHIT | | | | |
| | | 15:8 | | | | | EOSRDY | CHNERRC | FLTRDY | CHRDYC |
| | | 23:16 | CHRDY[7:0] | | | | | | | |
| | | 31:24 | CHRDY[15:8] | | | | | | | |
| 0x0128 | INTFLAG | 7:0 | SOVFL | | | CMPHIT | CMPINTID[3:0] | | | |
| | | 15:8 | CRDYID[3:0] | | | | EOSRDY | CHNERRC | FLTRDY | CHRDYC |
| | | 23:16 | CHRDY[7:0] | | | | | | | |
| | | 31:24 | CHRDY[15:8] | | | | | | | |
| 0x012C ... 0x0167 | Reserved | | | | | | | | | |
| 0x0168 | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.1. CONTROL A REGISTER

**Name:** CTRLA
**Offset:** 0x000
**Reset:** 0x00000080
**Property:** RW

**Table 36-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | ONDEMAND | RUNSTDBY | | AIPMPEN | | ANAEN | ENABLE | SWRST |
| Access | R/W | R/W | | R/W | | R/W | R/W | R/W |
| Reset | 1 | 0 | | 0 | | 0 | 0 | 0 |

**Bit 7 – ONDEMAND** On Demand Control

The On Demand operation mode allows the ADC to be normally disabled, lowest power mode, until enabled by an EVSYS trigger event. The consequences of this however is that the analog logic is completely powered down when ONDEMAND=1 and clocks are suspended. Upon a EVSYS trigger event, it will require up to 20-50µs for the ADC to warm up meaning that the actual sample/convert sequence will not happen for:
ADC analog input signal sample point in time = (EVSYS Trigger event + CTRLD.WKUPEXP)
Which can be up to 50µs after the initial EVSYS trigger event. After the conversion is complete, if there are no pending EVSYS trigger events the ADC will again power down and the sequence will repeat as before.
When ONDEMAND = 1, the ADC works only on a rising edge EVSYS events. In addition, all other ADC trigger sources other than EVSYS triggers will not be acknowledged and therefore will render the ADC inactive. (See table below)
In Idle and Standby mode, if ONDEMAND = 1 and CTRLA.RUNSTDBY = 1, the ADC will be active and not powered down. If CTRLA.RUNSTDBY=0, the ADC is powered down and ADC clocks are suspended.
**Note:** CTRLD.WKUPEXP should be set by user to the equivalent of 50 µs or CTRLD.WKUPEXP[xxx] = 50 µs / TAD.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 36-7.** ONDEMAND Matrix

| SYSTEM STATUS | ONDEMAND | RUNSTDBY | ADC Trigger Source | ADC BEHAVIOR |
|---|---|---|---|---|
| ACTIVE | 1 | x | ADC Global Software Trigger. | Do Nothing |
| | | x | ADC Global Level Software Trigger | Do Nothing |
| | | x | ADC SCANTRG | Do Nothing except only when SCAN trigger source selected is EVSYS trigger, then RUN only when EVSYS trigger request is received. |
| | | x | ADC STRIG Synchronous Trigger | Do Nothing |
| | | x | ADC Trigger Event(s) from Event System (EVSYS) | RUN only when EVSYS trigger request is received. |
| | 0 | x | ADC Global Software Trigger. | ADC active, Trigger sample/conv and wait for next trigger |
| | | x | ADC Global Level Software Trigger | ADC active, Trigger sample/conv continuously. |
| | | x | ADC SCANTRG | ADC active, Trigger sample/conv and wait for next trigger |
| | | x | ADC STRIG Synchronous Trigger | ADC active, Trigger sample/conv and wait for next trigger |
| | | x | ADC Trigger Event(s) from Event System (EVSYS) | ADC active, Trigger sample/conv and wait for next trigger. |
| IDLE, STANDBY | 0 | 1 | All triggers working | ADC active |
| | 1 | 0 | n/a | Do nothing |
| | 1 | 1 | ADC Trigger Event(s) from Event System (EVSYS) | RUN only when EVSYS trigger request is received. |

| Value | Description |
|---|---|
| 0 | The ADC is always on, if ENABLE = 1. (Highly Recommended) |
| 1 | See table below. |

**Bit 6 – RUNSTDBY** Run in Standby

This bit controls how the ADC behaves during Standby Sleep mode.
This bit is not synchronized.

| Value | Description |
|---|---|
| 0 | Discontinue module operation when device enters idle/standby mode (but do not stop the CTL_CLK to preserve the analog biasing). The ADC will complete the ongoing conversion before disabling the module. |
| 1 | The ADC is not stopped in Idle or 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 4 – AIPMPEN** Charge Pump Enable

| Value | Description |
|---|---|
| 0 | Analog input charge pump is disabled. |
| 1 | Analog input charge pump is enabled. |

**Bit 2 – ANAEN** Analog Enable

**Note:** This bit must be set prior to setting any of the CTRLD.ANLENn bits.

| Value | Description |
|---|---|
| 0 | The analog and bias circuitry for the ADC module is powered down and clocks disabled. Trigger events will not be serviced. |
| 1 | Enables and powers up the ADC module analog logic based on the settings of CTRLD.ANLEN0 = 1. Whenever the ADC module exits a power down state to a powered-up state, CTRLA.ANAEN = 1 and CTRLD.ANLEN0 = 1, then the system hardware will wait until CTRLD.WKUPEXP time has expired before starting an ADC sample/conversion if a trigger event is pending. The ADC required warm-up time is 20-50us before it will stabilize for a valid ADC sample/conversion sequence to start. CTRLA.ENABLE does not need to be set to power-up the analog logic and begin the CTRLD.WKUPEXP time out. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 1 – ENABLE** ENABLE BIT

**Notes:**

1. The ENABLE bit should be set only after the ADC module has been configured. Changing the configuration bits after enabling the ADC could result in unpredictable behavior.
2. When ENABLE = 0 the internal control logic is reset, and all status generated by the module is cleared. All ADC Module clocks are disabled unless ANAEN = 1. In this case ANLEN0 = 0 disable clocks and ANLEN0 = 1 enables them. All ADC registers are available for reading and writing.

| Value | Description |
|---|---|
| 0 | A/D converter is off. |
| 1 | A/D converter module is enabled. |

**Bit 0 – SWRST** SOFTWARE RESET BIT

Writing a one to the SWRST bit resets the state of the ADC and all the registers to their initial state. The only exception is the DBGCTRL.DBGRUN bit, which will keep its value after a SWRST. The module will be disabled after the reset (ENABLE = 0). When writing a one to SWRST, no other bits in the same register will be written, as SWRST will clear all the bits in the same register.
After writing a one to SWRST, SWRST will read back one until the module and the registers are reset. Any register write access during the ongoing reset will be discarded and an error will be generated. Read access can be performed without an error generated and must return reset value. Writing a one to SWRST will have priority above all other actions and will always happen immediately.
Writing a zero to SWRST has no effect.
**Note:** During a SWRST, access to registers/bits without SWRST are disallowed until 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.

## 36.7.2. CONTROL B REGISTER

**Name:** CTRLB
**Offset:** 0x004
**Reset:** 0x00000000
**Property:** RW

**Table 36-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | SWCNVEN | | | | STRGEN | TRGSUSP | LSWTRG | GSWTRG |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | SAMP | RQCNVRT | | | ADCHSEL[3:0] | | | |
| Access | R/W | R/W | | | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | | | 0 | 0 | 0 | 0 |

**Bit 15 – SWCNVEN** Software Conversion Enable

**Note:** When CTRLB.SWCNVEN=1, the channels sample and conversions are controlled exclusively by SAMP and RQCNVRT. All hardware triggers and GSWTRG and LSWTRG are disabled. It will place the ADC module and all of its channels under the exclusive control of the CTRLB.SAMP and CTRLB.RQCNVRT bits.

| Value | Description |
|---|---|
| 0 | Software controlled conversions disable. Traditional ADC hardware triggers as defined by CHNCFG4 and CHNCFG5 will be active if enabled. |
| 1 | SAMP and RQCNVRT bits control entire ADC module sample and convert respectively for all channels. Setting this bit blocks all other hardware triggers events defined by CHNCFG4 and CHNCFG5. |

**Bit 11 – STRGEN** Synchronous Trigger Enable

**Note:** This bit is ignored if CTRLB.SWCNVEN=1.

| Value | Description |
|---|---|
| 0 | Disable automatic ADC hardware CTRLC.CNT counter driven synchronous ADC triggers. |
| 1 | Enable automated hardware synchronous trigger period defined by CTRLC.CNT register. |

**Bit 10 – TRGSUSP** Trigger Suspend


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | Trigger suspend disabled. Triggers if enabled and selected will occur. |
| 1 | Blocks triggers from starting new ADC conversions but does not disable the ADC Modules or disable trigger capture, (i.e. persistent last trigger event is latched). If trigger capture during trigger suspend is not desired for any channels, then CTRLD.CHNEN0 must be cleared then set in that order prior to resetting TRGSUSP. Pulsing down and up CTRLD.CHNEN0 prior to resetting TRGSUSP will clear ALL pending triggers. |

**Bit 9 – LSWTRG** Level Global Trigger

**Notes:**

1. This bit is ignored if SWCNVEN=1.
2. If SWCNVEN=0, this bit is NOT self-clearing and is meant to allow the user software to implement continuous sample/conversions on the associated analog input channel.

| Value | Description |
|---|---|
| 0 | Global Level Software Trigger disabled. |
| 1 | Trigger A/D conversions for ADC analog input(s) "y" that have selected LSWTRG bit as the trigger signal via the associated CHNCFG4.TRGSRC[y] or CHNCFG5.TRGSRC[y] = 0b0010 or via the CORCTRL.STRGSRC. |

**Bit 8 – GSWTRG** Global Software Trigger

This software settable bit will trigger ADC sample/conversion sequences for ADC inputs that have selected the GSWTRG bit as the trigger signal via the associated CHNCFG4.TRGSRC[y] or CHNCFG5.TRGSRC[y], via the CORCTRL.STRGSRC value. This bit is auto cleared on the next APB clock cycle and is meant to implement single conversions on trigger edge-sensitive channels.

**Note:**

1. This bit is ignored if SWCNVEN=1.

| Value | Description |
|---|---|
| 0 | Disable Global Software Trigger. |
| 1 | If SWCNVEN=0, this bit is auto cleared by hardware after sampling cycle has been triggered and is meant to implement single sample/conversions on trigger edge-sensitive channels. |

**Bit 7 – SAMP** Channel Sample

**Notes:**

1. This bit is ignored if SWCNVEN = 0.
2. ADCHSEL[3:0] must be initialized at or before when the SAMP bit is set.
3. The SAMP bit will keep the S&H circuit in Sample mode until the bit is cleared by the users software. Also, usage of the SAMP bit will cause settings of respective CORCTRL.SAMC<9:0> bits to be ignored.
4. The SAMP bit is not a self-clearing bit and it is the responsibility of application software to clear this bit but only after setting the RQCNVRT bit to start the analog-to-digital conversion.
5. When the SAMP and RQCNVRT bits are used by software routines, all TRGSRCx<4:0> bits and STRGSRC<4:0> bits should be set to '00000' to disable all hardware triggers and prevent them from interfering with the software-controlled sampling command signal SAMP and with the software controlled trigger RQCNVRT.
6. The SAMP bit should only be used in conjunction with RQCNVRT for user software-controlled sampling and triggering.

| Value | Description |
|---|---|
| 0 | ADC is not software-controlled. Sampling the channel selected by ADCHSEL[3:0]. |
| 1 | ADC is sampling and remain sampling for as long as user has this bit set in SW. The sampled analog input is defined by ADCHSEL[3:0] provided SWCNVEN = 1. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 6 – RQCNVRT** Request Channel Convert

**Notes:**

1. If user's software sets RQCNVRT = 1, an ADC conversion will begin immediately and terminate the sampling period defined by CORCTRL.SAMC. This bit is cleared by hardware after the conversion is complete.
2. If SWCNVEN = 1, after setting this bit the users software must immediately clear the SAMP bit.
3. This bit is ignored if SWCNVEN = 0, (i.e. if software-controlled sample/conversions are disabled).

| Value | Description |
|---|---|
| 0 | ADC is not converting if SWCNVEN = 1 or user previously set this bit and the previous ADC conversion is complete for the analog channel that was defined by ADCHSEL[3:0] |
| 1 | Terminate sampling and begin conversion of ADC and analog channel defined by ADCHSEL[3:0]. This bit is cleared by hardware when the conversion is complete. |

**Bits 3:0 – ADCHSEL[3:0]** Software Trigger Channel Select

This binary encoded bit field selects the ADC analog input to be sampled and converted respectively by the SAMP and RQCNVRT bit if SWCNVEN = 1.

**Note:** These bits are ignored if SWCNVEN = 0.

| ADCHSEL[3:0] | Analog Channel Input |
|---|---|
| 0x0 | AIN0 |
| 0x1 | AIN1 |
| 0x2 | AIN2 |
| 0x3 | AIN3 |
| 0x4 | AIN4 |
| 0x5 | AIN5 |
| 0X6 | AIN6 |
| 0x7 | AIN7 |
| 0x8 | AIN8 |
| 0x9 | AIN9 |
| 0xA | AIN10 |
| 0xB | AIN11 |
| 0xC | VDDCORE_SW |
| 0xD | Reserved |
| 0xE | 1.2V IVREF |
| 0xF | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.3. Control C Register

**Name:** CTRLC
**Offset:** 0x008
**Reset:** 0x00000000
**Property:** RW

**Table 36-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | CNT[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 |
|---|---|---|---|---|---|---|---|---|
| | CNT[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 – CNT[15:0]** Clock Divider for Synchronous Trigger
Free-running counter based on CTL_CLK times out when it reaches this value. At time out, the STRIG synchronous trigger will fire.
**Notes:**

1. This register is not valid unless either [CORCTRLn.STRGSRC=0x4 plus CHNCFG40/50.TRGSRCx=0x3 plus CHNCFG20.CSSx=1] or [CHNCFG40/50.TRGSRCx=0x4 plus CTRLB.SWCNVEN=0] for Synchronous Trigger from CTRLC.CNT.
2. CTL_CLK = GCLK_ADC / (CTRLD.CTLCKDIV+1)
3. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1 and return a bus error).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.4. Control D Register

**Name:** CTRLD
**Offset:** 0x010
**Reset:** 0x00000000
**Property:** RW

**Table 36-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | VREFSEL[2:0] | | | WKUPEXP[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | ANLEN0 | | | | CHNEN0 |
| Access | | | | R/W | | | | R/W |
| Reset | | | | 0 | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | CTLCKDIV[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

**Bits 30:28 – VREFSEL[2:0]** Voltage Reference Select

| VREFSEL[2:0] | ADREF+ | ADREF- |
|---|---|---|
| 000 | $A_{VDD}$ | $A_{VSS}$ |
| 001 | External $V_{REF}H$ | $A_{VSS}$ |
| 010-111 | Reserved | Reserved |

**Note:**
1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

| Value | Name | Description |
|---|---|---|
| 0 | AVDD_AVSS | AVDD and AVSS |
| 1 | EXTERNAL_VREFH_AVSS | External VREFH and AVSS |

**Bits 27:24 – WKUPEXP[3:0]** Wakeup cycles
These bits represent the number of ADC TAD clocks required to warm-up the ADC module before it can perform conversion.
1111 = 215 = 32,768 TAD clocks
* * *
* * *
0110 = 26 = 64 TAD clocks


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

0101 = 25 = 32 TAD clocks
0100 = 24 = 16 TAD clocks
0011 = 24 = 16 TAD clocks
0010 = 24 = 16 TAD clocks
0001 = 24 = 16 TAD clocks
0000 = 24 = 16 TAD clocks

**Notes:**

1. Minimum required ADC warm-up time, (i.e., WKUPCLKCNT), is 50 µs. (i.e., TAD * WKUPCLKCNT).
2. TAD = [1 / ((GCLK_ADC Freq / (CTRLD.CTLCKDIV+1)) / (2 * CORCTRL.ADCDIV)) ] or (GCLK_ADC Period * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV).
3. These bits are Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

**Bit 20 – ANLENx** Analog and Bias circuitry Enable for Core x

**Notes:**

1. The CALCTRL register must be initialized by the users software to the factory-provided values in the Calibration Configuration Register FCCFG65 before setting ANLENx=1.
2. CTRLA.ANAEN=1 must be set prior to setting ANLENx=1.
3. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.

| Value | Description |
|---|---|
| 0 | ADC Analog and bias circuitry powered down and clocks suspended. Analog logic power saving mode. |
| 1 | Analog and bias circuitry enabled. Once the analog and bias circuit is enabled CTRLA.ANLEN=1 and ANLEN, the ADC module needs a warm-up time, as defined by the WKUPEXP<3:0> bits. |

**Bit 16 – CHNENx** Core x Digital Channel Enable

**Notes:**

1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and return a bus error.
2. ANLENx qualifies CHNENx: If ANLENx =0, then digital logic is also disabled.

| Value | Description |
|---|---|
| 0 | ADC digital logic disabled. No trigger, sample or conversion events will be processed. (power-saving mode with fast 2 TAD clock wakeup provided ANLEN0 = 1). |
| 1 | ADC digital logic enabled (required for active operation). |

**Bits 13:8 – CTLCKDIV[5:0]** Control Clock Divider

The CTLCKDIV bit field divides the GCLK_ADC input clock into the ADC Module control clock CTL_CLK scaled by CTRLD.CTLCKDIV with period $T_Q$.

**Note:** This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1 and returns a bus error.

| Value | Description |
|---|---|
| 111111 | $T_{GCLK_ADCx}$·(CTLCKDIV[5:0]+1) = 64·$T_{GCLK_ADCx}$ = $T_Q$ |
| 111110 | $T_{GCLK_ADCx}$·(CTLCKDIV[5:0]+1) = 63·$T_{GCLK_ADCx}$ = $T_Q$ |
| . . . | ... |
| 000100 | $T_{GCLK_ADCx}$·(CTLCKDIV[5:0]+1) = 5·$T_{GCLK_ADCx}$ = $T_Q$ |
| 000011 | $T_{GCLK_ADCx}$·(CTLCKDIV[5:0]+1) = 4·$T_{GCLK_ADCx}$ = $T_Q$ |
| 000010 | $T_{GCLK_ADCx}$·(CTLCKDIV[5:0]+1) = 3·$T_{GCLK_ADCx}$ = $T_Q$ |
| 000001 | $T_{GCLK_ADCx}$·(CTLCKDIV[5:0]+1) = 2·$T_{GCLK_ADCx}$ = $T_Q$ |
| 000000 | $T_{GCLK_ADCx}$·(CTLCKDIV[5:0]+1) = 1·$T_{GCLK_ADCx}$ = $T_Q$ |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.5. SARCORE Control

**Name:** CORCTRL
**Offset:** 0x020
**Reset:** 0x00000C00
**Property:** RW

**Table 36-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | ADCDIV[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 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | SCNRTDS | STRGLVL | | STRGSRC[3:0] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | EIS[2:0] | | | SELRES[1:0] | | SAMC[9:8] | |
| Access | | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SAMC[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 30:24 – ADCDIV[6:0]** Division Ratio for SARCORE clock
ADCDIV provides the ADC analog functional clock referred to as the ADC clock or TAD, (i.e., the period of the ADC clock). It divides CTL_CLK output clock from (GCLK_ADC Period * (CTRLD.CTLCKDIV+1)) with period = TQ to produce the resulting ADC_CLK for ADC sampling and converting. It will have the period TAD given by the formula:
TAD = (GCLK_ADC Period * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV)
**Note:** The ADC throughput rate $F_{TPR}$ = [1 / ((SAMC+2) * TAD) + ((#bits of resolution selected+1)*TAD)] / # of active AINx inputs.

**Notes:**

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1) . Returns a bus error.
2. The minimum sample rate for ADC0 is Ftpr >= 100 ksps.
3. For optimal performance and accuracy, the user must select the maximum TAD clock supported by the device as specified in the ADC electrical specs:
   TAD = (GCLK_ADCx Period * (CTRLD.CTLCKDIV+1)) * (2 * ADCDIV)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

4. If the user wishes to control the ADC throughput rate, they must do so by modulating either CORCTRLn.SAMC or the EVSYS trigger interval or both. Do not try to do so by reducing the TAD clock frequency as this will increase the ADC conversion time allowing the charge on the ADC holding capacitor from the sampled analog input signal to leak off, attenuating the ADC result accuracy.

| Value | Description |
|---|---|
| 1111111 | $254 \cdot T_Q = T_{AD}$ |
| ... | ... |
| 0000011 | $6 \cdot T_Q = T_{AD}$ |
| 0000010 | $4 \cdot T_Q = T_{AD}$ |
| 0000001 | $2 \cdot T_Q = T_{AD}$ |
| 0000000 | Reserved |

**Bit 22 – SCNRTDS** SCAN Re-trigger Disable

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

| Value | Description |
|---|---|
| 0 | Allows the scan cycle to restart from the beginning, lowest CSSn channel even before all of the scan channels have been measured if a new scan trigger arrives before then. If this happens then the INTENCLR.EOSRDY flag will be set when the current scan is interrupted, and the new scan starts from the beginning of the lowest CSSn channel selected. |
| 1 | Prevents an early scan trigger (arriving before of the end of the current scan) from starting a new scan cycle. The scan will include all channels which have their CHNCFG4.TRGSRCk set to point to the Scan Trigger, i.e., CHNCFG4.TRGSRCy[3:0] = 4'b0011, and have their associated CHNCFG2.CSSk bit set. |

**Bit 21 – STRGLVL** Scan Trigger Level Sensitivity

**Note:**

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

STRGLVL functions as follows:

| Value | Description |
|---|---|
| 0 | SCANTRG is Positive Edge Active(the power-up value for backwards compatibility).<br>A positive edge on the SCANTRG will initiate a single but complete scan of all included channels. |
| 1 | SCANTRG is High Level Active. So long as SCANTRG stays high, the entire scan will re-trigger. |

**Bits 19:16 – STRGSRC[3:0]** SCAN trigger source selection

These bits select the trigger source for the scan trigger SCANTRG. The trigger STRIG serves all the channels k, which have their CHNCFG{4|5}.TRGSRCk = 4'h0011, and have their Channel Scan Select bit set (CHNCFG2.CSSk = 1'b1).

**Notes:**

1. This bit is Enabled Protected: Writes are ignored when CTRLA.ENABLE = 1 returns a bus error.
2. In order to utilize CORCTRLn.STRGSRC=0x4, user must configure the following registers:
   a. CTRLC.CNT.
   b. CHNCFG2n.CSSx.
   c. CHNCFG4/5n.TRGSRCx=0x3.
   d. CTRLB.SWCNVEN = 0.

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 14:12 – EIS[2:0]** Early Interrupt Select

These bits select the number of core clocks and TAD clocks prior to the end of conversion at which the early interrupt is generated. All channels share the same EIS setting.
The interrupt is generated ((EIS +1) x TAD) ADC Module clocks prior to end of conversion.

**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) . returns a bus error.
2. Early interrupt is a feature that can be useful in ADC very high speed data acquisitions to mask the MCU interrupt latency delay.
   Example: CPU MCLK =72 Mhz, ADC CTL_CLK = 36 MHz and ADC TAD Clk = 18 MHz.
   Hypothetically, if it took the CPU 20 MCLK cycles to service an interrupt, then theoretically the ADC early interrupt EIS could be set to 4, hence the ISR latency was masked. The CPU could then read the ADC result almost immediately from within the Interrupt Service Routine (ISR), when the result was ready without any concern for overwriting the current result or FIFO with the next ADC conversion result.

**Notes:** Depending on the bit resolution selection field SELRES, the allowed maximum EIS values are: as follows

- 12-bit Resolution, all 8 possible settings are allowed, 0 - 7
- 10-bit Resolution, all 8 possible settings are allowed, 0 - 7
- 8-bit Resolution, only the 6 lowest settings are allowed, 0 - 5
- 6-bit Resolution, only the 4 lowest settings are allowed, 0 - 3

The hardware will utilize the maximum allowed EIS setting if the user sets the EIS value that is too big.

**⚠ CAUTION** Do not set the EIS to large or the interrupt can occur before the ADC conversion is complete and the CPU could read the previous ADC conversion result instead of the one in progress.

| Value | Description |
|---|---|
| 0x0 | Generate interrupt 1 TAD clock before end of ADC conversion |
| 0x1 | Generate interrupt 2 TAD clocks before end of ADC conversion |
| 0x2 | Generate interrupt 3 TAD clocks before end of ADC conversion |
| 0x3 | Generate interrupt 4 TAD clocks before end of ADC conversion |
| 0x4 | Generate interrupt 5 TAD clocks before end of ADC conversion |
| 0x5 | Generate interrupt 6 TAD clocks before end of ADC conversion |
| 0x6 | Generate interrupt 7 TAD clocks before end of ADC conversion |
| 0x7 | Generate interrupt 8 TAD clocks before end of ADC conversion |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 11:10 – SELRES[1:0]** Selects Resolution

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1). Returns a bus error.

| Value | Name | Description |
|---|---|---|
| 0 | 6_BITS | 6 bits |
| 1 | 8_BITS | 8 bits |
| 2 | 10_BITS | 10 bits |
| 3 | 12_BITS | 12 bits (power-on default) |

**Bits 9:0 – SAMC[9:0]** Sample Count

The sample time required depends on the external analog signal source impedance. (See ADC electrical characteristics of SAMC values required based on external source impedance. If the external analog source impedance is unknown or if your getting inconsistent ADC results consider increasing the SAMC sample time at the cost of a lower ADC throughput rate of course.
ADC Throughput Rate
$F_{TPR}$ = [1 / ((SAMC value +2) * TAD) + ((#bits of resolution selected+1)*TAD)] / # of active AINx inputs

**Notes:**

1. All channels share the same SAMC setting.
2. The sampling sequence starts with a trigger event.
3. The internal sampling cap is not discharged between samples.
4. These bits are Enabled Protected: Writes are ignored when CTRLA.ENABLE = 1 returns a bus error.
5. When IVREF is to be sampled (SAMC value + 2) * TAD must be ≧ 30 us.

| Value | Description |
|---|---|
| 1111111111 | 1025 $T_{AD}$n |
| ... | ... |
| 0000000001 | 3 $T_{AD}$n |
| 0000000000 | Reserved |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.6. Channel Configuration 1 (LVL/CMPEN)

**Name:** CHNCFG1
**Offset:** 0x024
**Reset:** 0x00000000
**Property:** RW

**Table 36-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | LVL | LVL | LVL | LVL | LVL | LVL | LVL | LVL |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | LVL | LVL | LVL | LVL | LVL | LVL | LVL | LVL |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN | CHNCMPEN |
| Access | R/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 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – LVL** Channel Level
Channel x is sensitive to the high level of its trigger. Otherwise it is sensitive only to the positive edge of its trigger.

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – CHNCMPEN** Channel Comparator Enable
Channel x is being monitored by the Digital Comparator which is servicing this channel. Otherwise, this channel will not be monitored by the Digital Comparator.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.7. Channel Configuration 2(FRACT/CSS)

**Name:** CHNCFG2
**Offset:** 0x028
**Reset:** 0x00000000
**Property:** RW

**Table 36-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT | FRACT |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | CSS | CSS | CSS | CSS | CSS | CSS | CSS | CSS |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | CSS | CSS | CSS | CSS | CSS | CSS | CSS | CSS |
| Access | R/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 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – FRACT** Fractional Data Output Format for Channel x

See *16-bit Output Data Format* table for format values.

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – CSS** Channel SCAN Select

When this bit is set, channel x is included in the scan cycle. Otherwise, this channel will not be included in the scan cycle.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.8. Channel Configuration3 (SIGN/DIFF)

**Name:** CHNCFG3
**Offset:** 0x02C
**Reset:** 0x00000000
**Property:** RW

**Table 36-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN | SIGN |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF | DIFF |
| Access | R/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 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – SIGN** SIGN setting
When set to 1, the associated channel x will have its output converted data in signed format. Otherwise, the output data will be in unsigned format.

**Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 – DIFF** Differential Mode
When set, the associated channel x is connected in differential mode. Otherwise, this channel is connected in single-ended mode.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.9. Channel Configuration 4 (TRGSRC)

**Name:** CHNCFG4
**Offset:** 0x030
**Reset:** 0x00000000
**Property:** RW

These bits select the conversion trigger source for the given channel x.

**Table 36-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC7[3:0] | | | | TRGSRC6[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 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC5[3:0] | | | | TRGSRC4[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 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC3[3:0] | | | | TRGSRC2[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 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC1[3:0] | | | | TRGSRC0[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 |

**Bits 31:28 – TRGSRC7[3:0]** Channel 7 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 27:24 – TRGSRC6[3:0]** Channel 6 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 23:20 – TRGSRC5[3:0]** Channel 5 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 19:16 – TRGSRC4[3:0]** Channel 4 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 15:12 – TRGSRC3[3:0]** Channel 3 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 11:8 – TRGSRC2[3:0]** Channel 2 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 7:4 – TRGSRC1[3:0]** Channel 1 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 3:0 – TRGSRC0[3:0]** Channel 0 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.10. Channel Configuration 5 (TRGSRC)

**Name:** CHNCFG5
**Offset:** 0x034
**Reset:** 0x00000000
**Property:** RW

These bits select the conversion trigger source for the given channel x.

**Table 36-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC15[3:0] | | | | TRGSRC14[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 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC13[3:0] | | | | TRGSRC12[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 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC11[3:0] | | | | TRGSRC10[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 |
|---|---|---|---|---|---|---|---|---|
| | TRGSRC9[3:0] | | | | TRGSRC8[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 |

**Bits 31:28 – TRGSRC15[3:0]** Channel 15 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 27:24 – TRGSRC14[3:0]** Channel 14 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 23:20 – TRGSRC13[3:0]** Channel 13 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 19:16 – TRGSRC12[3:0]** Channel 12 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 15:12 – TRGSRC11[3:0]** Channel 11 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 11:8 – TRGSRC10[3:0]** Channel 10 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 7:4 – TRGSRC9[3:0]** Channel 9 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |

**Bits 3:0 – TRGSRC8[3:0]** Channel 8 Trigger Source

| Value | Name | Description |
|---|---|---|
| 0 | NO_TRIGGER | No Trigger (NOP) |
| 1 | GLOBAL_SOFTWARE_TRIGGER | Global Software Trigger |
| 2 | GLOBAL_LEVEL_TRIGGER | Global Level Software Trigger |
| 3 | SCAN_TRIGGER | SCANTRG - Scan Trigger |
| 4 | SYNC_TRIGGER | Synchronous Trigger (STRIG) |
| 5 | EVENT_USER0 | ADC Trigger Event User 0 |
| 6 | EVENT_USER1 | ADC Trigger Event User 1 |
| 7 | EVENT_USER2 | ADC Trigger Event User 2 |
| 8 | EVENT_USER3 | ADC Trigger Event User 3 |
| 9 | EVENT_USER4 | ADC Trigger Event User 4 |
| 10 | EVENT_USER5 | ADC Trigger Event User 5 |
| 11 | EVENT_USER6 | ADC Trigger Event User 6 |
| 12 | EVENT_USER7 | ADC Trigger Event User 7 |
| 13 | EVENT_USER8 | ADC Trigger Event User 8 |
| 14 | EVENT_USER9 | ADC Trigger Event User 9 |
| 15 | EVENT_USER10 | ADC Trigger Event User 10 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.11. SARCORE Calibration Value

**Name:** CALCTRL
**Offset:** 0x038
**Reset:** 0x00000000
**Property:** RW

**Table 36-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CALBITS[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 |
|---|---|---|---|---|---|---|---|---|
| | CALBITS[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 |
|---|---|---|---|---|---|---|---|---|
| | CALBITS[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 |
|---|---|---|---|---|---|---|---|---|
| | CALBITS[7:0] | | | | | | | |
| Access | RW | RW | RW | RW | RW | RW | RW | RW |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:0 – CALBITS[31:0]** Calibration Values

These register bits can only change when the applicable CTRLD.ANLEN0 bit is 0. Once the applicable CTRLD.ANLENn bit rising edge occurs, those bit values must remain unchanged until after the same CTRLD.ANLEN0 bit falling edge occurs.
This register must be initialized in user software to the factory-provided values in the Calibration Configuration Register FCCFG65 before setting CTRLD.ANLENn to 1.

**Note:**

1. This bit is Enabled Protected (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 36.7.12. Event Control

**Name:** EVCTRL
**Offset:** 0x03C
**Reset:** 0x00000000
**Property:** RW

**Table 36-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | CMPEO | RESRDYEO | STARTINV | | | STARTEI |
| Access | | | R/W | R/W | R/W | | | R/W |
| Reset | | | 0 | 0 | 0 | | | 0 |

**Bit 5 – CMPEO** Comparator Window Event Out

This bit indicates whether the Digital Comparator Window event output is enabled or not and whether an output event will be generated when the Digital Comparator Window is met.

**Note:**

1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.

| Value | Description |
|---|---|
| 0 | Window event output is disabled, and an event will not be generated. |
| 1 | Window 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.

**Note:**

1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.

| 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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 3 – STARTINV** Start Conversion Invert

**Notes:**

1. Only rising edge EVSYS ADC Trigger events are acknowledged.
2. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.

| Value | Description |
|---|---|
| 0 | Start event input source is not inverted. |
| 1 | Start event input source is inverted. |

**Bit 0 – STARTEI** Start Event Conversion Input Enable

**Note:**

1. This bit is Enabled Protected. Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.

| Value | Description |
|---|---|
| 0 | Not enabled. ADC Event System events cannot trigger start of conversions. |
| 1 | Enabled. ADC Event System events can trigger start of conversions. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.13. Comparator Control

**Name:** CMPCTRL
**Offset:** 0x0B0
**Reset:** 0x00000000
**Property:** RW

**Table 36-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | IEHIHI | IEHILO | ADCMPHI[11:8] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADCMPHI[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 |
|---|---|---|---|---|---|---|---|---|
| | IEBTWN | IELOHI | IELOLO | CMPEN | ADCMPLO[11:8] | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | ADCMPLO[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 29 – IEHIHI** Enable VAL >= CMPHI
Setting this bit enables comparison events ADCMPHI≤ ADC value.
**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are ‘0, therefore in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0’s.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

**Bit 28 – IEHILO** Enable VAL < CMPHI
Setting this bit enables comparison events ADC value < ADCMPHI.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

**Bits 27:16 – ADCMPHI[11:0]** High Limit of Digital Comparator

This register stores the limit value which is used for comparisons with the ADC Module output data when
IEHIHI = 1, IEHILO = 1, or IEBTWN = 1.
The user is responsible for formatting the data in ADCMPHI[11:0] as signed or unsigned to match the data format as specified by the CHNCFG3.SIGNk and CHNCFG2.FRACTk bits for all the analog input channels k which are enabled by CHNCFG1.CHNCMPENk .
**Note:** In Filter accumulation mode, the comparison is done on the upper 12 of the 16 bits of filter data.

**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

**Bit 15 – IEBTWN** Enable CMPLO <= VAL < CMPHI

Setting this bit enables comparison events ADCMPLO ≤ ADC Value < ADCMPHI.
**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

**Bit 14 – IELOHI** Enable VAL >= CMPLO

Setting this bit enables comparison events ADCMPLO≤ ADC Value.
**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

| Value | Description |
|---|---|
| 0 | Use normal interrupts |
| 1 | use early interrupts |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 13 – IELOLO** Enable VAL < CMPLO

Setting this bit enables comparison events ADC Value < ADCMPLO.

**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

**Bit 12 – CMPEN** Comparator Enable

Setting this bit enables digital comparisons for the inputs to the ADC.
For each channel input channel k to be compared the corresponding bit CHNCFG1.CHNCMPENk must be set for the channel to be monitored.

**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.

**Bits 11:0 – ADCMPLO[11:0]** Low Limit of Digital Comparator

This register stores the limit value which is used for comparisons with the ADC Module output data when
IELOHI = 1, IELOLO = 1, or IEBTWN = 1.
The user is responsible for formatting the data in ADCMPLO[11:0] as signed or unsigned to match the data format as specified by the CHNCFG3.SIGNk and CHNCFG2.FRACTk bits for all the analog input channels k which are enabled by CHNCFG1.CHNCMPENk .
**Note:** In Filter accumulation mode, the comparison is done on the upper 12 of the 16 bits of filter data.

**Notes:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.
2. The Digital Comparator works on the final value of the filter data.
3. When using channel using FRACT (16bits left justified) the lower 4 bits are '0, therefor in this mode 15:4 will be in 11:0 and user must account for the lower 4 bits of 0's.
4. In any case where the results value is of greater resolution than 12 bits, the comparison is only performed on upper 12 bits of the results value according to the settings of register CMPCTRL.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.14. Filter Control

**Name:** FLTCTRL
**Offset:** 0x0C0
**Reset:** 0x00000000
**Property:** RW

**Table 36-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | FLTCHNID[3:0] | | | | | FLTEN |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | DATA16EN | FMODE | OVRSAM[2:0] | | |
| Access | | | | R/W | R/W | R/W | R/W | R/W |
| Reset | | | | 0 | 0 | 0 | 0 | 0 |

**Bits 13:10 – FLTCHNID[3:0]** Channel ID
Identifies which input channel, k , is to be filtered by the Digital Filter.
**Note:**
1. This bit is Enabled Protected. (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

**Bit 8 – FLTEN** Filter Enable
When set, this bit enables the Digital Filter to filter the output data generated by the ADC. The input channel to be filtered is determined by FLTCHNID.
**Note:**
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

**Bit 4 – DATA16EN** 16bit Averaging Mode
**Note:**
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

This bit is significant only if FMODE = 1 (Averaging Mode) and CHNCFG2.FRACTk= 1 (Fractional Output Mode, where k = FLTCHNID[3:0] is the chosen input for filtering) as follows:

| Value | Description |
|---|---|
| 0 | Only the first 12 bits are significant, followed by 4 zeros. |
| 1 | All 16 bits of the filter output data are significant |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 3 – FMODE** Filter Mode

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

| Value | Description |
|---|---|
| 0 | Filtering in Oversampling Mode (power-up default) |
| 1 | Filtering in Averaging Mode |

**Bits 2:0 – OVRSAM[2:0]** Oversampling Ratio

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1) returns a bus error.

Determines the number of samples generated in the burst mode used for computing one single filter output value.
The OVRSAM encoding depends on the FMODE setting as follows:
If FMODE = 1 (Oversampling Mode) then OVRSAM is encoded as follows:

| Value | Descripton |
|---|---|
| 000 | 4 samples, shift sum 1-bit to right, output data is 13-bits |
| 001 | 16 samples, shift sum 2-bits to right, output data is 14-bits |
| 010 | 64 samples, shift sum 3-bits to right, output data is 15-bits |
| 011 | 256 samples, shift sum 4-bits to right, output data is 16-bits |
| 100 | 2 samples, shift sum 0-bits to right, output data is in 12.1 format |
| 101 | 8 samples, shift sum 1-bit to right, output data is in 13.1 format |
| 110 | 32 samples, shift sum 2-bits to right, output data is in 14.1 format |
| 111 | 128 samples, shift sum 3-bits to right, output data is in 15.1 format |

If FMODE=0 (Averaging Mode), then OVRSAM is encoded as follows:

| Value | Descripton |
|---|---|
| 000 | 2 samples to be averaged |
| 001 | 4 samples to be averaged |
| 010 | 8 samples to be averaged |
| 011 | 16 samples to be averaged |
| 100 | 32 samples to be averaged |
| 101 | 64 samples to be averaged |
| 110 | 128 samples to be averaged |
| 111 | 256 samples to be averaged |

| Value | Name | Description |
|---|---|---|
| 0 | 4_SAMPLES | (If FMODE is 0) 4 samples (shift sum 1 bit to right, output data is 13-bits) / (If FMODE is 1) 2 samples to be averaged |
| 1 | 16_SAMPLES | (If FMODE is 0) 16 samples (shift sum 2 bits to right, output data is 14-bits) / (If FMODE is 1) 4 samples to be averaged |
| 2 | 64_SAMPLES | (If FMODE is 0) 64 samples (shift sum 3 bits to right, output data is 15-bits) / (If FMODE is 1) 8 samples to be averaged |
| 3 | 256_SAMPLES | (If FMODE is 0) 256 samples (shift sum 4 bits to right, output data is 16-bits) / (If FMODE is 1) 16 samples to be averaged |
| 4 | 2_SAMPLES | (If FMODE is 0) 2 samples (shift sum 0 bits to right, output data is 12.1 format) / (If FMODE is 1) 32 samples to be averaged |
| 5 | 8_SAMPLES | (If FMODE is 0) 8 samples (shift sum 1 bit to right, output data is 13.1 format) / (If FMODE is 1) 64 samples to be averaged |
| 6 | 32_SAMPLES | (If FMODE is 0) 32 samples (shift sum 2 bits to right, output data is 14.1 format) / (If FMODE is 1) 128 samples to be averaged |
| 7 | 128_SAMPLES | (If FMODE is 0) 128 samples (shift sum 3 bits to right, output data is 15.1 format) / (If FMODE is 1) 256 samples to be averaged |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.15. Channel Ready DATA ID

**Name:** CORCHDATAID
**Offset:** 0x0D0
**Reset:** 0x00000000
**Property:** RW

**Table 36-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | CORDYID[1:0] | | CHRDYID[3:0] | | | |
| Access | | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | | | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 5:4 – CORDYID[1:0]** Core Read ID
This bit field is always read as '0'

**Bits 3:0 – CHRDYID[3:0]** Channel Read ID
Input Channel Index k:
For ADC, set value to input channel index, for status register CHNRDYDAT to display the current values of configuration bits and the last converted output data or written by the user to display the channel.
**Note:**

1. Selecting unimplemented input channels on a given ADC will return a bus error with the data (32'h00000000).

| Value | Description |
|---|---|
| 1110 | analog input channel 14 |
| ... | ... |
| 0010 | analog input channel 2 |
| 0001 | analog channel 1 |
| 0000 | analog channel 0 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.16. Channel Ready Data Register

**Name:** CHRDYDAT
**Offset:** 0x0D4
**Reset:** 0x00000000
**Property:** RW

**Note:** All the data read in this register pertains to the Module defined by CORCHDATAID.CORDYID and its analog input channel defined by CORCHDATAID.CHRDYID.

**Table 36-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | FRACT | SIGN | DIFF | LVL |
| Access | | | | | R/K | R/K | R/K | R/K |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CHRDYDAT[15:8] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CHRDYDAT[7:0] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit 27 – FRACT** Fractional Setting

**Bit 26 – SIGN** Sign Setting

**Bit 25 – DIFF** Differential Setting

**Bit 24 – LVL** Level Setting

**Bits 15:0 – CHRDYDAT[15:0]** Channel Output Data

**Note:** A read of CHRDYDAT will generate a read bus error on analog input channels which have not been implemented and the returned data will be 0x0000_0000.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.17. APB FIFO Output Data

**Name:** PFFDATA
**Offset:** 0x0D8
**Reset:** 0x00000000
**Property:** R

**Note:** Reading any part of the PFFDATA register advances the FIFO pointer, therefore it is recommended to read all 32 bits of the register and then parse out the values of each field.

**Table 36-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | PFFCNT[7:0] | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | PFFFRACT | PFFSIGN | PFFCORID[1:0] | | PFFCHNID[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 |
|---|---|---|---|---|---|---|---|---|
| | PFFDATA[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 |
|---|---|---|---|---|---|---|---|---|
| | PFFDATA[7:0] | | | | | | | |
| Access | R | R | R | R | R | R | R | R |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bits 31:24 – PFFCNT[7:0]** Current Data Entries in APB FIFO
Current number of data entries in the APB FIFO

**Bit 23 – PFFFRACT** Fractional Setting from APB FIFO
ADC SAR Core FRACTx setting associated with the data in the PFFDATA register.

**Bit 22 – PFFSIGN** Channel Sign from the APB FIFO
Channel sign setting associated with data in the PFFDATA register.

**Bits 21:20 – PFFCORID[1:0]** Core ID from APB FIFO
This bit always reads 0

**Bits 19:16 – PFFCHNID[3:0]** Channel ID from APB FIFO

| Value | Description |
|---|---|
| 000 | Channel 1 |
| 0001 | Channel 2 |
| 0010 | Channel 3 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0011 | Channel 4 |
| 0100 | Channel 5 |
| 0101 | Channel 6 |
| 0110 | Channel 7 |
| 0111 | Channel 8 |
| 1000 | Channel 9 |
| 1001 | Channel 10 |
| 1010 | Channel 11 |
| 1011 | Channel 12 |
| 1100 | Channel 13 |
| 1101 | Channel 14 |
| 1110 | Channel 15 |
| 1111 | Channel 16 |

**Bits 15:0 – PFFDATA[15:0]** SARCORE Conversion data from the APB FIFO
Data is in the format given by PFFFRACT and PFFSIGN.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.18. APB FIFO Control Register

**Name:** PFFCTRL
**Offset:** 0x0E4
**Reset:** 0x00000000
**Property:** RW

The ADC FIFO is useful in applications that stream out ADC data at very high transfer rates to relive CPU bandwidth. Individual high data rate ADC result interrupts and CPU reads may slow bus access transfer requests.

**Table 36-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | PFFRDYDM |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | PFFCR0 | | | PFFEN | |
| Access | | | | R/W | | | R/W | |
| Reset | | | | 0 | | | 0 | |

**Bit 16 – PFFRDYDM** DMA APB FIFO Data Ready

| Value | Name | Description |
|---|---|---|
| 0 | CTLINTFLAG_PFFHFUL | Selects CTLINTFLAG.PFFHFUL for the trigger signal to the DMAC |
| 1 | CTLINTFLAG_PFFRDY | Selects CTLINTFLAG.PFFRDY for the trigger signal to the DMAC |

**Bit 4 – PFFCR0** APB CORE 0 FIFO Enable
When PFFEN = 1, setting this bit enables the conversion output data of any channel k to be stored into the optional data FIFO.
**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

**Bit 1 – PFFEN** APB FIFO Enable
When the FIFO is disabled no data is being saved into the FIFO and the its logic is being kept in reset state.
**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Description |
|---|---|
| 0 | FIFO is disabled |
| 1 | FIFO is enabled |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.19. CORE SYNC Busy Status Register

**Name:** SYNCBUSY
**Offset:** 0x0E8
**Reset:** 0x00000000
**Property:** R

**Notes:** The following GCLK-clocked registers are “Enable Write Protected”:

- CTRLC
- CTRLD
- CORCTRL
- CHNCFG1
- CHNCFG2
- CHNCFG3
- CHNCFG4
- CHNCFG5
- CALTRL
- FLTCTRL

Therefore, they do not require a SyncBusy bit. These registers are write-disabled when the CTRLA.ENABLE bit is set to enable the ADC. They can only be changed when the ADC is disabled (CTRLA.ENABLE = 0). The user must completely configure the ADC and then enable the ADC by setting CTRLA.ENABLE = 1.

**Table 36-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | CTRLB | ENABLE | SWRST |
| Access | | | | | | R | R | R |
| Reset | | | | | | 0 | 0 | 0 |

**Bit 2 – CTRLB** CTRLB sync busy
For the GCLK-based register CTRLB:
When Hardware sets this bit, no writes are permitted to the CTRLB register.

**Bit 1 – ENABLE** Enable bit Sync Busy
For GCLK-based register bit CTRLA.ENABLE:
When Hardware sets this bit, no writes are permitted to the CTRLA.ENABLE register.

**Bit 0 – SWRST** Software Reset Sync Busy
**Notes:**

1. Typically, when the SWRST is written, the bit is auto-cleared the next APB clock cycle after. However, the SYNCBUSY.SWRST bit is set and stays set until the reset in the GCLK domain is completed. Therefore, the user must poll the SYNCBUSY register to know when the operation is complete.
2. Care must be taken during the APB reset phase, because potentially the external clock (GCLK) may not present.
3. During a SWRST, access to registers or bits without SWRST are disallowed until SYNCBUSY.SWRST is cleared by hardware.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.20. CORE Controller Interrupt Enable Set

**Name:** CTLINTENSET
**Offset:** 0x0FC
**Reset:** 0x00000000
**Property:** RW

**Note:** Writing a zero to these bits has no effect. Writing a one to these any of these bits will SET the ENABLE bit.

**Note:** A read of this register provides whether the Interrupt is Enabled (bit=1) or Disabled (bit=0), i.e. a write of a 1 to the bit then a read of the bit will return the interrupt is enabled (=1).

**Table 36-26.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PFFHFUL | PFFRDY | PFFOVF | PFFUNF |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VREFRDY | VREFUPD | | | | | | CRRDY0 |
| Access | R/W | R/W | | | | | | R/W |
| Reset | 0 | 0 | | | | | | 0 |

**Bit 11 – PFFHFUL** APB FIFO Half Full
Writing a 1 to this bit will enable the ADC FIFO Half Full interrupt request.
Reading this bit returns whether the PFFHFUL interrupt is enabled (1 = enabled).

**Bit 10 – PFFRDY** APB FIFO Ready
Writing a 1 to this bit will enable the ADC FIFO Data Ready interrupt request.
Reading this bit returns whether the PFFRDY interrupt is enabled (1 = enabled).

**Bit 9 – PFFOVF** APB FIFO Overflow
Writing a 1 to this bit will enable the ADC FIFO Overflow Error interrupt request,
Reading this bit returns whether the PFFOVF interrupt is enabled (1 = enabled).

**Bit 8 – PFFUNF** APB FIFO Underflow
Writing a 1 to this bit will enable the ADC FIFO Read Underflow Error interrupt request.
Reading this bit returns whether the PFFUNF interrupt is enabled (1 = enabled).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 7 – VREFRDY** VREF Ready
Writing a 1 to this bit will Enable the ADC Voltage Reference Ready as an interrupt request.
Reading this bit returns whether the VREFRDY interrupt is enabled (1 = enabled).

**Bit 6 – VREFUPD** VREF update
Writing a 1 to this bit will Enable the ADC Voltage Reference Ready Updated as an interrupt request.
Reading this bit returns whether the VREFUPD interrupt is enabled (1 = enabled).

**Bit 0 – CRRDYx** Core x Ready
Writing a 1 to this bit will enable the Core x (where x is up to 0) Ready as an interrupt request.
Reading this bit returns whether the CRRDYx interrupt is enabled (1 = enabled).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.21. CORE Controller Interrupt Enable Clear

**Name:** CTLINTENCLR
**Offset:** 0x100
**Reset:** 0x00000000
**Property:** RW

**Note:** Writing a zero to these bits has no effect. Writing a one to these any of these bits will CLEAR the ENABLE bit.

**Note:** A read of this register provides whether the Interrupt is Enabled (bit=1) or Disabled (bit=0), i.e. a write of a 1 to the bit then a read of the bit will return the interrupt is disabled (=0).

**Table 36-27.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PFFHFUL | PFFRDY | PFFOVF | PFFUNF |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VREFRDY | VREFUPD | | | | | | CRRDY0 |
| Access | R/W | R/W | | | | | | R/W |
| Reset | 0 | 0 | | | | | | 0 |

**Bit 11 – PFFHFUL** APB FIFO Half Full
Writing a 1 to this bit will disable the ADC FIFO Half Full as an interrupt request.
Reading this bit returns whether the PFFHFUL interrupt is enabled (1 = enabled).

**Bit 10 – PFFRDY** APB FIFO Ready
Writing a 1 to this bit will disable the ADC FIFO Data Ready as an interrupt request.
Reading this bit returns whether the PFFRDY interrupt is enabled (1 = enabled).

**Bit 9 – PFFOVF** APB FIFO overflow
Writing a 1 to this bit will disable the ADC FIFO Overflow Error as an interrupt request,
Reading this bit returns whether the PFFOVF interrupt is enabled (1 = enabled).

**Bit 8 – PFFUNF** APB FIFO underflow
Writing a 1 to this bit will disable the ADC FIFO Read Underflow Error as an interrupt request.
Reading this bit returns whether the PFFUNF interrupt is enabled (1 = enabled).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 7 – VREFRDY** VREF Ready
Writing a 1 to this bit will disable the ADC Voltage Reference Ready as an interrupt request.
Reading this bit returns whether the VREFRDY interrupt is enabled (1 = enabled).

**Bit 6 – VREFUPD** VREF Update
Writing a 1 to this bit will disable the ADC Voltage Reference Ready Updated as an interrupt request.
Reading this bit returns whether the VREFUPD interrupt is enabled (1 = enabled).

**Bit 0 – CRRDYx** Core Ready x Disable
Writing a 1 to this bit will disable the Core x (where x is up to 0) Ready as an interrupt request.
Reading this bit returns whether the CRRDYx interrupt is enabled (1 = enabled).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.22. CORE Controller Interrupt Flags

**Name:** CTLINTFLAG
**Offset:** 0x104
**Reset:** 0x00000000
**Property:** RW

**Table 36-28.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | PFFHFUL | PFFRDY | PFFOVF | PFFUNF |
| Access | | | | | R | R | R/K | R/K |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | VREFRDY | VREFUPD | | | | | | CRRDY0 |
| Access | R | R/K | | | | | | R |
| Reset | 0 | 0 | | | | | | 0 |

**Bit 11 – PFFHFUL** APB FIFO Half Full

This bit is set when the FIFO is at least half full of data to be read. When set, this bit enables this condition to trigger the ADC interrupt if the corresponding bit in CTLINTSET is set. This bit is cleared by hardware when the FIFO output data has been read and there is less than a half full FIFO left to be read.
This bit is NOT reset by software writing a 1 to it, it is only reset by hardware.

**Bit 10 – PFFRDY** APB FIFO Ready

This bit is set when the FIFO has data to be read. When set, this bit enables the trigger of the ADC interrupt if the corresponding bit in CTLINTSET is set.
This bit is cleared by hardware when the FIFO output data in has been read and there is no additional data ready in the FIFO (that is the APB FIFO is empty). This bit is NOT reset by software writing a 1 to it.

**Bit 9 – PFFOVF** APB FIFO overflow

This bit is set by hardware when the FIFO is full and new output data overwrites data to be read. When set, this bit enables this condition to trigger an ADC interrupt if the corresponding bit in CTLINTSET is set.
This bit is reset by software writing a 1 to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 8 – PFFUNF** APB FIFO underflow

This bit is set by hardware when the FIFO is empty. When set, this bit enables the trigger of an ADC interrupt if the corresponding bit in CTLINTSET is set.
This bit is reset by software writing a 1 to it.

**Bit 7 – VREFRDY** VREF Ready

Hardware sets and clears this bit according to the status of the ADC Voltage Reference. ADC Conversion Data is valid only after VREFRDY is set by hardware, so the ADC interrupt service routine in charge of data processing should always check first that VREFRDY is set to ensure the data validity. This bit will only be updated to the VREFRDY condition set or cleared when the CTRLA.ENABLE is on. Therefore, once the ISR has verified that VREFRDY=1, it should disable the corresponding interrupt by setting CTLINTENCLR.VREFRDY = 1 to prevent continuous firing of the ISR. Alternately, the software setup routine for the ADC can wait in a while(1) loop, polling VREFRDY until it goes high.

**Bit 6 – VREFUPD** VREF update

This bit is set by hardware on both the positive and negative edges of the bit VREFRDY. This means that the hardware will set this bit when the ADC Voltage Reference is ready, but also when it fails, that is on any change. When set, this bit enables the trigger of an ADC interrupt if the corresponding bit in CTLINTSET is set.
This bit is reset by software writing a 1 to it. It is NOT cleared by a software read.
Software must read the value of VREFRDY to ascertain if the ADC analog reference circuits are in order or not when the CPU is servicing the interrupt prompted by VREFUPD. The ADC Voltage Reference analog signals are required to be ready during operation of the ADC. If an ADC Voltage Reference fault is detected, the ADC module must be re-calibrated. Most likely an ADC Reference Voltage fault is caused by a brown-out of the analog Vdd supply.
Hardware sets this bit to zero when CTRLA.ENABLE=0.

**Bit 0 – CRRDYx** Core x Ready

This bit is set by hardware when a $2^{WKUPEXP}$ number of ADC SAR Core clocks have lapsed after software setting CTRLD.ANLENx to one. It is cleared by hardware when CTRLD.ANLENx is de-asserted (but does NOT depend on CTRLD.CHNENx). It is not cleared after a software read. The value of CRRDYx, if set, enables the CRRDYx as part of a group interrupt request


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.23. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x120
**Reset:** 0x00000000
**Property:** RW

**Note:** Writing a zero to any of these bits has no effect, but writing a one to these bits will CLEAR the ENABLE bit.
A read of this register provides the current status of interrupts, that is whether each interrupt is enabled (bit = 1) or disabled (bit = 0).

**Table 36-29.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CHRDY[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 |
|---|---|---|---|---|---|---|---|---|
| | CHRDY[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | EOSRDY | CHNERRC | FLTRDY | CHRDYC |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SOVFL | | | CMPHIT | | | | |
| Access | R/W | | | R/W | | | | |
| Reset | 0 | | | 0 | | | | |

**Bits 31:16 – CHRDY[15:0]** Channel Ready Disable
Writing a 1 to bit will k disable the data ready flag for channel k as an interrupt request.
**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

**Bit 11 – EOSRDY** Endo of Scan Disable
Writing a 1 to this bit will disable the flag bit EOSRDY as an interrupt request.
**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error).

**Bit 10 – CHNERRC** Channel Overwrite Error Disable
Writing a 1 to this bit will disable the flag bit CHNERRCas an interrupt request.
**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 9 – FLTRDY** Filter Ready Disable

Writing a 1 to this bit will disable the filter ready flag for Filter as an interrupt request.

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 8 – CHRDYC** Core Current Channel Disable

Writing a 1 to this bit will disable the flag bit CHRDYC as an interrupt request.

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 7 – SOVFL** Synchonizer Overflow Disable

Writing a 1 to this bit will disable the flag SOVFL as an interrupt request.

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 4 – CMPHIT** Compare Hit Disable

Writing a 1 to this bit will disable the comparator hit flag for Comparator n as an interrupt request.

**Note:**

1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.24. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x124
**Reset:** 0x00000000
**Property:** RW

**Note:** Writing a zero to any of these bits has no effect, but writing a one to these bits will SET the ENABLE bit.
A read of this register provides the current status of interrupts, i.e., whether each interrupt is enabled (bit=1) or disabled (bit=0).

**Table 36-30.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CHRDY[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 |
|---|---|---|---|---|---|---|---|---|
| | CHRDY[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | EOSRDY | CHNERRC | FLTRDY | CHRDYC |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SOVFL | | | CMPHIT | | | | |
| Access | R/W | | | R/W | | | | |
| Reset | 0 | | | 0 | | | | |

**Bits 31:16 – CHRDY[15:0]** Channel Ready Enable
Writing a 1 to bit k will enable the data ready flag for Core n channel k as an interrupt request.
CHRDY[ k ] is defined only for k = 0,1,2,...,(Sn-1).
**Note:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 11 – EOSRDY** End of Scan Enable
Writing a 1 to this bit will disable the flag bit EOSRDY as an interrupt request.
**Note:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 10 – CHNERRC** Channel Overwrite Enable
Writing a 1 to this bit will enable the flag bit CHNERRC as an interrupt request.
**Note:**

1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 9 – FLTRDY** Filter Ready Enable
Writing a 1 to this bit will enable the filter ready flag for Filter as an interrupt request.
**Note:**
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 8 – CHRDYC** Current Channel Ready Enable
Writing a 1 to this bit will enable the flag bit CHRDYC as an interrupt request.
**Note:**
1. This bit is Enabled Protected : (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 7 – SOVFL** Synchronizer Overflow Enable
Writing a 1 to this bit will enable the flag SOVFL as an interrupt request.
**Note:**
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)

**Bit 4 – CMPHIT** Compare Hit Enable
Writing a 1 to this bit will enable the comparator hit flag for Comparator n as an interrupt request.
**Note:**
1. This bit is Enabled Protected: (Writes are ignored when CTRLA.ENABLE = 1. Returns a bus error.)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.25. Interrupt Flags

**Name:** INTFLAG
**Offset:** 0x128
**Reset:** 0x00000000
**Property:** RW

**Note:** Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

**Table 36-31.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CHRDY[15:8] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | CHRDY[7:0] | | | | | | | |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | CRDYID[3:0] | | | | EOSRDY | CHNERRC | FLTRDY | CHRDYC |
| Access | R/K | R/K | R/K | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SOVFL | | | CMPHIT | CMPINTID[3:0] | | | |
| Access | R/K | | | R/K | R/K | R/K | R/K | R/K |
| Reset | 0 | | | 0 | 0 | 0 | 0 | 0 |

**Bits 31:16 – CHRDY[15:0]** Channel Ready
CHRDY[ k ] = 1 indicates that ADC has completed its last A/D conversion for channel k.
**Note:** If INTENSET.CHRDY[ k ] is set, then CHRDY[ k ] = 1 will trigger an ADC interrupt.
CHRDY[ k ] is reset by software writing a 1 to it.

**Bits 15:12 – CRDYID[3:0]** Channel Ready ID
The value of CRDYID indicates the input channel index, k , that ADC has just completed in its current scan. This if for information only since these bits cannot be used to trigger an ADC interrupt. (CHRDY[ k ] are intended for that purpose.)
These bits are reset by software writing a 1 to them.

**Bit 11 – EOSRDY** End of Scan Ready
This bit is set by hardware at the end of the scan of all channels included in the scan performed by ADC in response to a SINGLE event of the Scan Trigger 0 (STRIG0).
If INTENSET.EOSRDY is set, then EOSRDY = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 10 – CHNERRC** Channel Overwrite Error
When set, this bit indicates that the ADC has completed its last A/D conversion for channel CRDYID[3:0], but at the time CRDYID[3:0] was updated, the status bit CHRDYC was still set, which indicates that the software may not have had the time to read the previous data, which may be now lost.
If INTENSET.CHNERRC is set, then CHNERRC = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

**Bit 9 – FLTRDY** Filter Ready
When set, this bit indicates that the digital filter has issued a new output sample for the input channel defined by FLTCTRL.FLTCHNID.
If INTENSET.FLTRDY is set, then FLTRDY = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

**Bit 8 – CHRDYC** Current Channel Ready
0 = ADC busy or idle.
1 = When set, this bit signifies that the ADC has completed its current A/D conversion for the channel identified in CRDYID[3:0].
**Notes:**
1. If INTENSET.CHRDYC is set, then CHRDYC = 1 will trigger an ADC interrupt.
2. This bit is reset by software writing a 1 to it.

**Bit 7 – SOVFL** Synchronizer Overflow
When set this bit signifies the ADC data was lost due to a slow APB_CLK.
If INTENSET.SOVFL is set, then SOVFL = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

**Bit 4 – CMPHIT** Compare Hit
When set, this bit signifies that the Digital Comparator associated with ADC has issued a condition hit interrupt for channel identified in CMPINTID[5:0].
If INTENSET.CMPHIT is set, then CMPHIT = 1 will trigger an ADC interrupt.
This bit is reset by software writing a 1 to it.

**Bits 3:0 – CMPINTID[3:0]** Compare Channel ID
When set, this signifies that the ADC Channel ID for which the digital comparator has issued the condition hit interrupt (CMPHIT=1). These bits cannot be used as an interrupt request flag. This is for information only. (CMPHIT is intended for that purpose.)
This bit is reset by software writing a 1 to it.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 36.7.26. Debug Control Register

**Name:** DBGCTRL
**Offset:** 0x168
**Reset:** 0x00000000
**Property:** RW

**Table 36-32.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | R/W |
| Reset | | | | | | | | 0 |

**Bit 0 – DBGRUN** Debug Running State
This bit is not affected by software reset and should not be changed by software while the ADC is enabled.
**Note:** During debug operation with DBGRUN=0 the ADC captures input trigger events and on exit from the debug the ADC will (based on priority) perform conversions on the captured event or events.

| Value | Description |
|---|---|
| 0 | The ADC is halted when the CPU is halted in debug mode. |
| 1 | The ADC continues normal operation when the CPU is halted in debug mode. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 37. Analog Comparator (AC)

## 37.1. Overview

The Analog Comparator (AC) consists of up to 1 comparator pairs. 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. The two comparators can combine to support a window function discussed below.

Hysteresis is user selectable to achieve the optimal operation for each application.

By default, the input selection includes four shared analog port pins and several internal signals. The flexible internal architecture allows additional input selections. For details, refer to each chip-specific documentation. 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, called Comparator 0 (COMP0) and Comparator 1 (COMP1) . The pair can be set in window mode to compare a signal to a voltage range instead of a single voltage level.

**Important:** If AVDD < 2.5v, users must enable analog charge pumps in SUPC.VREGCTRL.CPEN[1:0].

## 37.2. Features

The following are key features of the AC module:

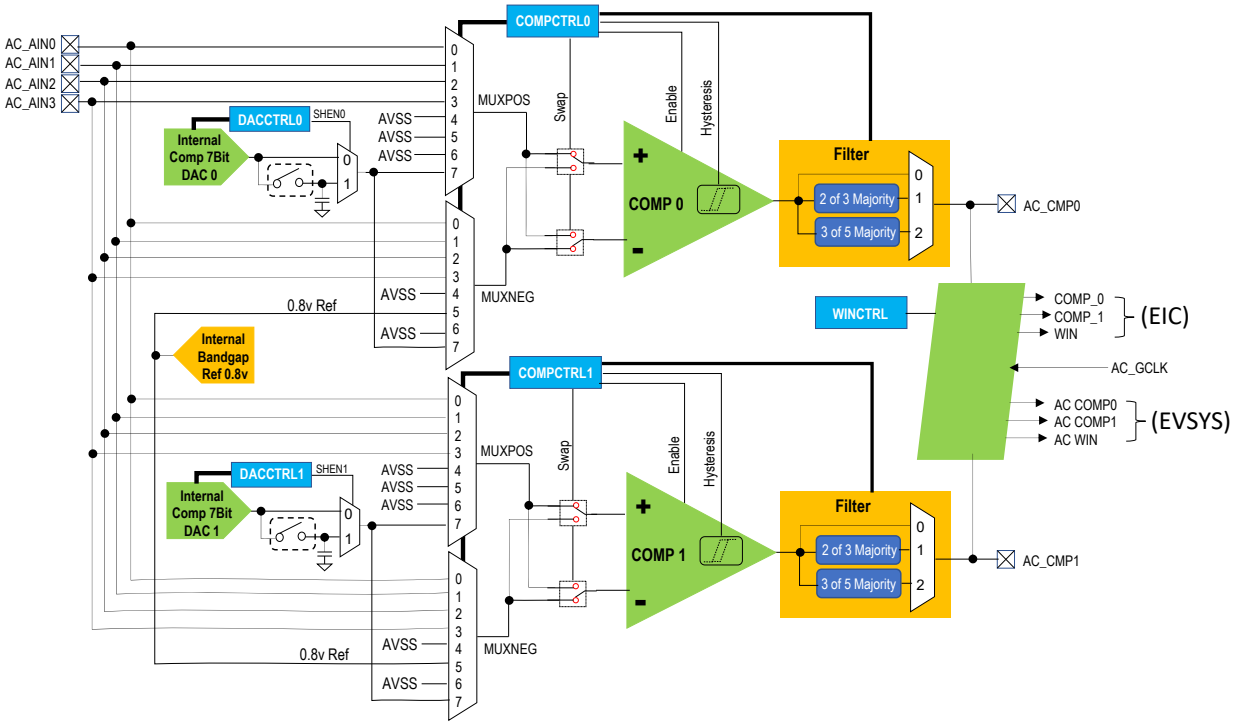
- 2 individual comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis
  - 4-levels, Always On
- Analog comparator outputs available on pins
  - Asynchronous or synchronous
- Default input selection:
  - 4 pins selectable for positive or negative inputs
  - Ground (for zero crossing)
  - Supports 7-bit dedicated DACs for each comparator, enabling up to 2^7-level programmable VDD scaler
  - Extensions available to connect additional internal or external pins on each device
- 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Event generation on:
  - Comparator output
  - Window function inside/outside window
- Optional digital filter on comparator output
- Low-power options:
  - Single-shot support
  - Continuous or sampling DAC operating mode
- Ability to delay the comparator output by n-cycle of controller clock

## 37.3. Block Diagram

**Figure 37-1.** Analog Comparator Block Diagram




## 37.4. Analog Connections

Each comparator has minimum four I/O pins that can be used as analog inputs. Each pair of comparators shares four pins. These pins must be configured for analog operation before using them as comparator inputs.

The Analog comparator embeds additional capabilities, enabling the option to connect more I/O pins from device to device, or other internal references. This selection is device specific, and for further details refer to the device specification.

Any internal reference source, such as a bandgap voltage reference or internal DAC must be configured and enabled prior to its use as a comparator input. Any external reference must be enabled and ready to be used, before the comparator is enabled.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 37.5. Functional Description

### 37.5.1. 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).

### 37.5.2. Basic Operation

#### 37.5.2.1. Initialization

Some registers are enable-protected, meaning they can only be written when the module is disabled.

The following register is enable-protected:

- The Control C register (CTRLC)
- The Event Control register (EVCTRL)
- The DAC n Control register (DACCTRLn)

Enable-protection is denoted by the 'Enable-Protected' property in each individual register description.

#### 37.5.2.2. 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.

#### 37.5.2.3. 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.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 Run in Standby bit (CTRLA.RUNSTDBY) and (COMPCTRLx.AC_RUNSTDBY)

The individual comparators are enabled by writing a '1' to the Enable bit in the COMPCTRL.ENABLEx. The individual comparators are disabled by writing a '0' to COMPCTRL.ENABLEx. Writing a '0' to CTRLA.ENABLE will also disable all the comparators, but will not clear their COMPCTRLx.ENABLE bits.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 37.5.2.4. 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 (tSTARTUP) 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. During the start-up time, the COMP output is not available. For details on how the start-up time is calculated, refer to COMPCTRLx.SUT description.

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.

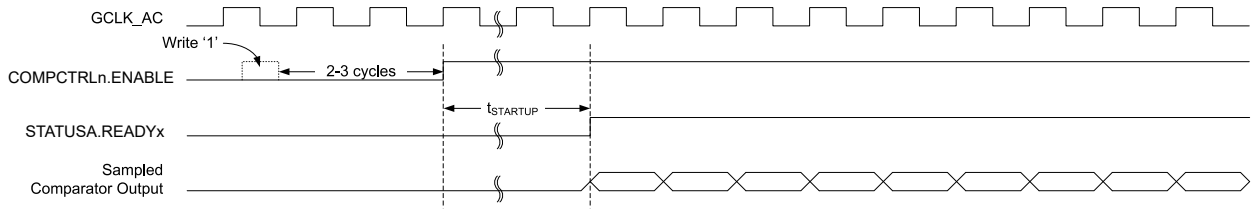
##### 37.5.2.4.1. Continuous Measurement

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 A register (STATUSA.READYx) is set, and the appropriate peripheral events and interrupts are also generated. New comparisons are performed continuously until the COMPENABLE.ENABLEx 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 following figure.

**Figure 37-2.** 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.

##### 37.5.2.4.2. 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.

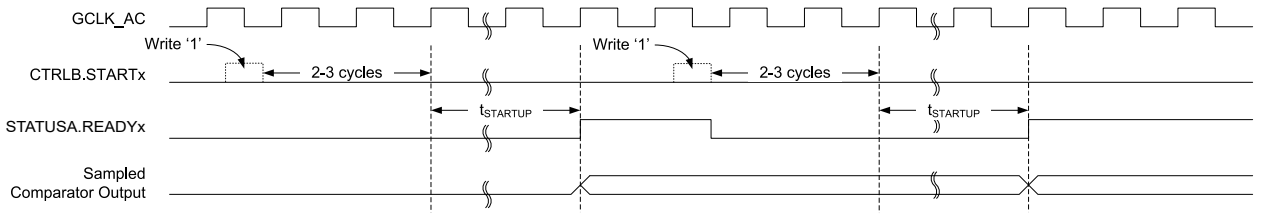

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Writing '1' to CTRLB.STARTx also clears the Comparator x Ready bit in the Status A register (STATUSA.READYx). STATUSA.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.

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 the following figure.

**Figure 37-3.** 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.

### 37.5.3. Selecting Comparator Inputs

Each comparator has one positive and one negative input. These inputs can come from one of the external input pins (AINx) or from one of the several internal reference voltage sources. 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.

### 37.5.4. 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).

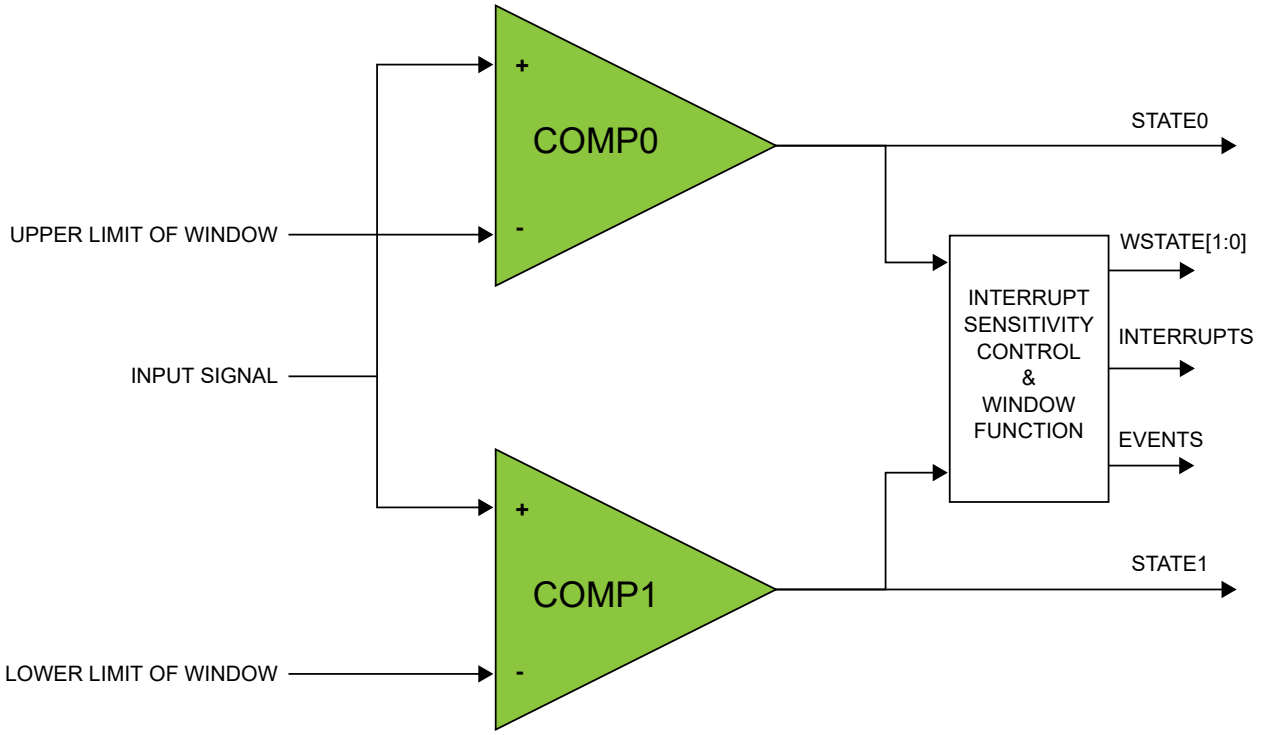
To physically configure the pair of comparators for Window mode, the same I/O pin must be chosen as positive input for each comparator, providing a shared input signal. The negative inputs define the range for the window. In the following figure, COMP0 defines the upper limit and COMP1 defines the lower 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).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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 when both comparators are ready. 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.

**Figure 37-4.** Comparators in Window Mode




### 37.5.5. Internal DAC Operation

Each Analog Comparator pair, includes two DACs, each connected to its respective ACx.

The DACx is enabled when the analog comparator is enabled (COMPCTRLn.ENABLE = 1), and the DAC is used as positive or negative input for the respective comparator (MUXNEG = INTDAC or MUXPOS = INTDAC).

The DAC is disabled when the analog comparator is disabled (COMPCTRLn.ENABLE = 0).

The DAC configuration registers are cleared when the analog comparator is reset (CTRLA.SWRST = 1).

The DAC generates a reference voltage that is a fraction of the device's supply voltage, with 2^7 levels. One independent DAC channel is dedicated for each comparator. The voltage of each channel is selected by the Value x bit field in the DACCTRL registers (DACCTRLn.VALUEx).

The DAC can be enabled in continuous operating mode (DACCTRLn.SHENx = 0), or in sample mode (DACCTRLn.SHENx = 1). When set in continuous mode, no clocks are required for operation. In sampling mode, each DAC controller includes a dedicated counter, generating the sampling clock.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

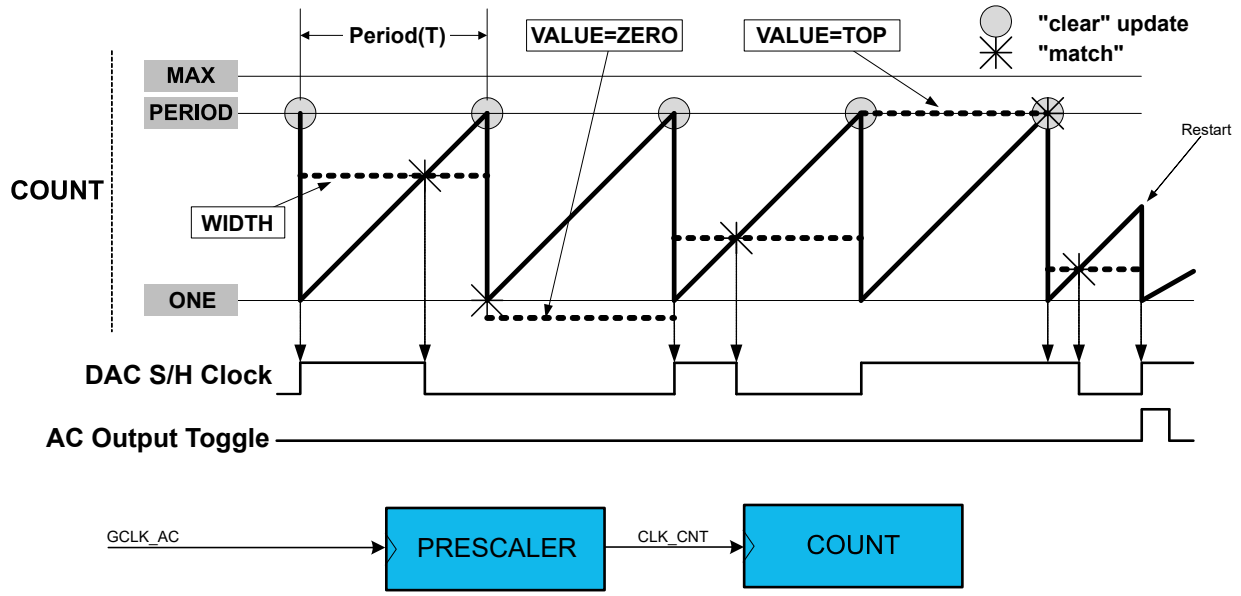
The counter operation is started when the analog comparator is ready (STATUSA.READYx = 1), and stopped when the analog comparator is disabled. As a consequence, in single shot mode, the DAC is enabled in continuous mode when the AC is enabled (COMPCTRLn.ENABLE = 1), and switches to the sample and hold operation when the comparator is ready.

**Note:** The DAC S/H operating mode and AC single-shot mode are two independent operations, and not linked together. Both S/H and continuous DAC operating modes can be selected when the AC is used in either continuous or single-shot mode. The operating modes must be selected by the application, depending on power consumption and/or response time requirements.

The counter counts up and restarts when the software PERIOD programmable value (CTRLC.PER) is reached. The counter operation is stopped and the internal counter is cleared when the sampling and hold operation is not required. When the AC output toggles, the counter is restarted.

The sampling clock is set when the counter operation starts, and cleared when the counter reaches the WIDTH software programmable value (CTRLC.WIDTH).

**Figure 37-5.** Sample and Hold Clock Generation




### 37.5.6. Input Hysteresis

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 always enabled for each comparator. The level of hysteresis is programmable through the Hysteresis Level bits also in the Comparator x Control register (COMPCTRLx.HYST).

### 37.5.7. Propagation Delay vs. Power Consumption

It is possible to trade off comparison speed for power efficiency to get the shortest possible propagation delay or the lowest power consumption. The speed setting is configured for each comparator individually by the Speed bit in the Comparator x Control register (COMPCTRLx.SPEED). The Speed bit selects the amount of bias current provided to the comparator, and as such will also affect the start-up time.

### 37.5.8. 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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:** 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 Continuous Mode Filtering. For single-shot mode, the comparison completes after the Nth filter sample, as shown in Single-Shot Filtering.

**Figure 37-6.** Continuous Mode Filtering

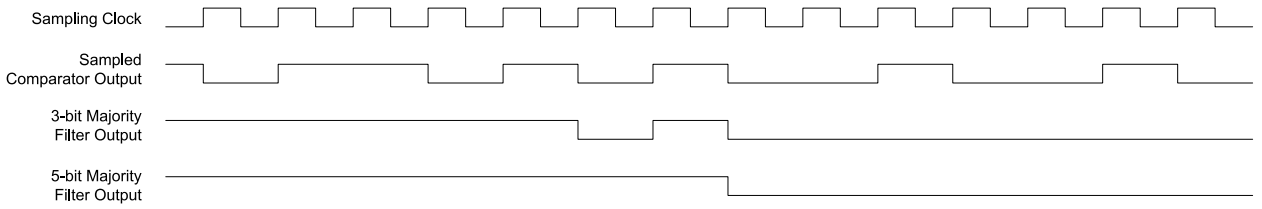
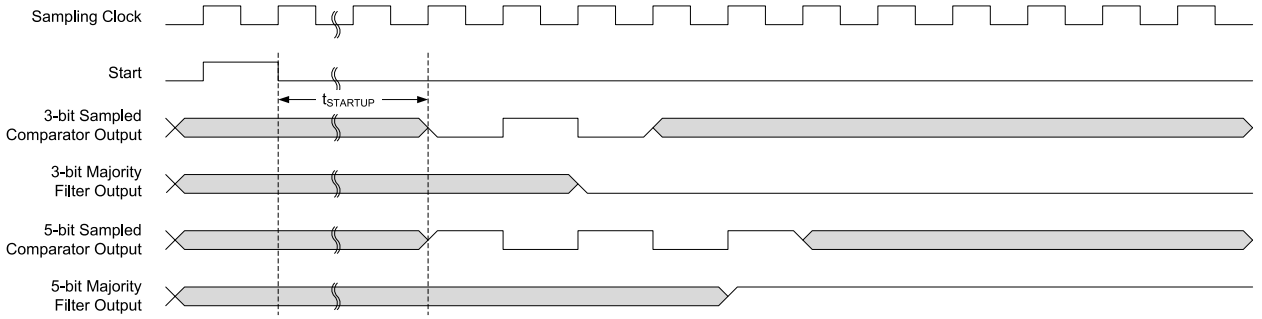



**Figure 37-7.** Single-Shot Filtering




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.

### 37.5.9. 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 n register (COMPCTRLn.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.

### 37.5.10. Interrupts

The AC has the following interrupt sources:

- Comparator (COMPx, x = {0,1}): Indicates a change in comparator status.
- 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 (COMPCTRLn.INTSEL). Window interrupts are generated based on the conditions selected by the Window Interrupt Selection bit group in the Window Control register (WINCTRL.WINTSELx).

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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

(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. Refer to the 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:** The interrupts must be globally enabled for interrupt requests to be generated.

### 37.5.11. Events

The AC can generate the following output events:

- Comparator (COMPx, x={0,(n+1)}): Generated as a copy of the comparator 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 (STARTx, x={0,1}): 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.

### 37.5.12. Sleep Mode Operation

The Run in Standby bits in the Comparator n Control registers (COMPCTRLn.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.

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 the following table.

**Table 37-1.** Sleep Mode Operation

| COMPCTRLn.MODE | RUNSTDBY=0 | RUNSTDBY=1 |
|---|---|---|
| 0 (Continuous) | GCLK_AC stopped, COMPx disabled | GCLK_AC stopped or running (depends on filtering options), COMPx enabled |
| 1 (Single-shot) | GCLK_AC stopped, COMPx disabled | GCLK_AC stopped, COMPx enabled only when triggered by an input event |

The Run in Standby bits in the Comparator n Control registers (COMPCTRLn.RUNSTDBY) also controls the behavior of the internal DAC during standby sleep mode, if the DAC is selected


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

a positive or negative multiplexor input (COMPCTRLn.MUXPOS = INTDAC, COMPCTRLn.MUXNEG = INTDAC). When the bit is zero, the DAC is disabled during sleep, but maintains its current configuration. When the bit is one, the DAC continues to operate during sleep. Note that when RUNSTDBY is zero, the DAC blocks are powered off for the lowest power consumption. This necessitates a start-up time delay when the system returns from sleep.

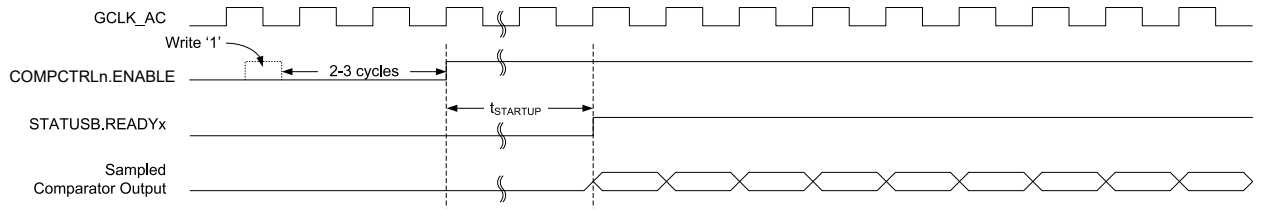
**Note:** When RUNSTDBY = 0 and sleep request is detected, COMPx remains active until the ongoing operation completes, and optionally interrupts can be set.

### 37.5.12.1. Continuous Measurement during Sleep

When a comparator is enabled in continuous measurement mode and the filtering options are disabled, the GCLK_AC is disabled during sleep and 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.

If the filtering is enabled, the GCLK_AC is kept running, the comparator will remain continuously enabled and will function synchronously. The current state of the comparator is synchronously monitored for changes. If an edge matching the interrupt condition is found and the interrupt is enabled in the Interrupt Enable registers (INTENCLR/SET), the AC can wake up the device.

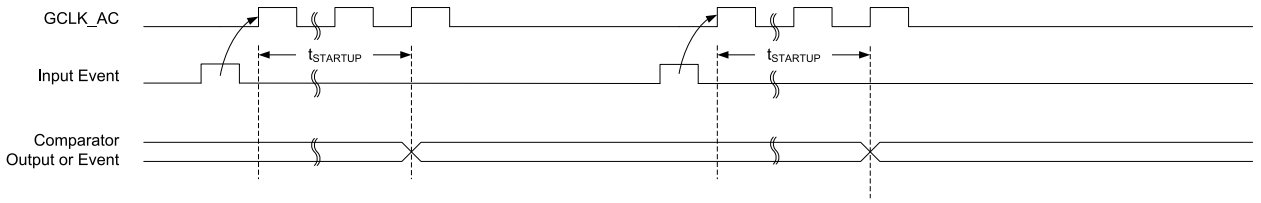
**Figure 37-8.** Continuous Mode SleepWalking




### 37.5.12.2. 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 the following figure. 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 37-9.** Single-Shot SleepWalking




### 37.5.13. 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:

- The Software Reset bit in the Control register (CTRLA.SWRST)
- The Enable bit in the Control register (CTRLA.ENABLE)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The Enable bit in the Comparator Control register (COMPCTRLn.ENABLE)

The following registers are synchronized when written:

- The Window Control register (WINCTRL)

Required write synchronization is denoted by the *Write-Synchronized* property in the register description.

## 37.6. Peripheral Dependencies

**Table 37-2.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| AC | 0x4482A000 | 132 : COMP0/ COMP1/WIN0 | **APB:** CLKMSK[1] MASK18 | PCHCTRL[34] | 40 STATUS1 PERID8 | | Generators: **COMP0:** CHANNEL[78] **COMP1:** CHANNEL[79] **WIN0:** CHANNEL[80] Users: **SOC0:** USER[49],Resynch **SOC1:** USER[50],Resynch | PD_CORE_SW |


© 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 | | | | | | | | |
| 0x04 | CTRLB | 7:0 | | | | | | | START1 | START0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x08 | CTRLC | 7:0 | WIDTH[7:0] | | | | | | | |
| | | 15:8 | PER[3:0] | | | | | | WIDTH[9:8] | |
| | | 23:16 | | | PER[9:4] | | | | | |
| | | 31:24 | | | | | AIPMPEN | PRESCALER[2:0] | | |
| 0x0C | EVCTRL | 7:0 | | | | | | | COMPEO1 | COMPEO0 |
| | | 15:8 | | | | | | | | WINEO0 |
| | | 23:16 | | | | | | | COMPEI1 | COMPEI0 |
| | | 31:24 | | | | | | | INVEI1 | INVEI0 |
| 0x10 | INTENCLR | 7:0 | | | | | | | COMP1 | COMP0 |
| | | 15:8 | | | | | | | | WIN0 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x14 | INTENSET | 7:0 | | | | | | | COMP1 | COMP0 |
| | | 15:8 | | | | | | | | WIN0 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x18 | INTFLAG | 7:0 | | | | | | | COMP1 | COMP0 |
| | | 15:8 | | | | | | | | WIN0 |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x1C | STATUSA | 7:0 | | | | | | | STATE1 | STATE0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | WSTATE0[7:0] | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x20 | STATUSB | 7:0 | | | | | | | READY1 | READY0 |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | DBGCTRL | 7:0 | | | | | | | | DBGRUN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | SYNCBUSY | 7:0 | | | | | COMPCTRL | COMPCTRL | ENABLE | SWRST |
| | | 15:8 | | | | | | WINCTRL | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x2C ... 0x2F | Reserved | | | | | | | | | |
| 0x30 | COMPCTRL0 | 7:0 | | RUNSTDBY | INTSEL[1:0] | | SINGLE | | ENABLE | |
| | | 15:8 | | MUXPOS[2:0] | | | | MUXNEG[2:0] | | |
| | | 23:16 | FLEN[2:0] | | | HYST[1:0] | | | SPEED | SWAP |
| | | 31:24 | SUT[5:0] | | | | | | OUT[1:0] | |
| 0x34 | COMPCTRL1 | 7:0 | | RUNSTDBY | INTSEL[1:0] | | SINGLE | | ENABLE | |
| | | 15:8 | | MUXPOS[2:0] | | | | MUXNEG[2:0] | | |
| | | 23:16 | FLEN[2:0] | | | HYST[1:0] | | | SPEED | SWAP |
| | | 31:24 | SUT[5:0] | | | | | | OUT[1:0] | |
| 0x38 | DACCTRL0 | 7:0 | | VALUE0[6:0] | | | | | | |
| | | 15:8 | SHEN0 | | | | | | | |
| | | 23:16 | | VALUE1[6:0] | | | | | | |
| | | 31:24 | SHEN1 | | | | | | | |


© 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 | DACCTRL1 | 7:0 | | | | | VALUE0[6:0] | | | |
| | | 15:8 | SHEN0 | | | | | | | |
| | | 23:16 | | | | | VALUE1[6:0] | | | |
| | | 31:24 | SHEN1 | | | | | | | |
| 0x3C | WINCTRL0 | 7:0 | | | | | | WINTSEL[1:0] | | WEN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x40 | WINCTRL1 | 7:0 | | | | | | WINTSEL[1:0] | | WEN |
| | | 15:8 | | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** RW

**Table 37-3.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 | | | | | | | RW | 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.

| 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.
**Note:** To avoid spurious interrupts from enable/disable cycles, use the SWRST bit to reset the comparator module.


© 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.

### 37.7.2. Control B

**Name:** CTRLB
**Offset:** 0x04
**Reset:** 0x00000000
**Property:** RW

**Table 37-4.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | START1 | START0 |
| Access | | | | | | | RW | RW |
| 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 it is not enabled in single-shot mode, Writing a '1' has no effect.
This bit always reads as zero.

| Value | Description |
|---|---|
| 0 | No effect |
| 1 | Starts a single-shot comparison on COMPx |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.3. Control C

**Name:** CTRLC
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** RW

**Table 37-5.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | AIPMPEN | PRESCALER[2:0] | | |
| Access | | | | | R/W | R/W | R/W | R/W |
| Reset | | | | | 0 | 0 | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | PER[9:4] | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | PER[3:0] | | | | | | WIDTH[9:8] | |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | WIDTH[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 27 – AIPMPEN** Analog Input Charge Pump Enable
This bit enables or disables the Analog Input Charge Pump

| Value | Description |
|---|---|
| 0x0 | The Analog Input Charge Pump is disabled |
| 0x1 | The Analog Input Charge Pump is enabled |

**Bits 26:24 – PRESCALER[2:0]** Prescaling Factor
These bits define the prescaling factor for the AC clock source (GCLK_AC) to generate the DAC sampling clock as shown in the following table.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV1 | Sampling rate is GCLK_AC (No division) |
| 0x1 | DIV2 | Sampling rate is GCLK_AC/2 |
| 0x2 | DIV4 | Sampling rate is GCLK_AC/4 |
| 0x3 | DIV8 | Sampling rate is GCLK_AC/8 |
| 0x4 | DIV16 | Sampling rate is GCLK_AC/16 |
| 0x5 | DIV32 | Sampling rate is GCLK_AC/32 |
| 0x6 | DIV64 | Sampling rate is GCLK_AC/64 |
| 0x7 | DIV128 | Sampling rate is GCLK_AC/128 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 21:12 – PER[9:0]** Sample and Hold Clock Period
These bits configure the sample and hold clock period. If PER is set to zero, no sample and hold DAC clock is generated.
**Note:** These bits are ignored if DACCTRLn.SHENn=0 (i.e. DAC continuous operation mode is enabled).

**Bits 9:0 – WIDTH[9:0]** Sample and Hold Clock Pulse Width
These bits configure the sample and hold clock pulse width. If WIDTH is set to zero, no sample and hold DAC clock is generated.
**Note:** These bits are ignored if DACCTRLn.SHENn=0 (i.e. DAC continuous operation mode is enabled).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.4. Event Control

**Name:** EVCTRL
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** RW

**Table 37-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | INVEI1 | INVEI0 |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | COMPEI1 | COMPEI0 |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | WINEO0 |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | COMPEO1 | COMPEO0 |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bits 24, 25 – 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 16, 17 – COMPEIx** Comparator x Event Input Enable
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 8 – WINEOx** Window x Event Output Enable
These bits indicate whether the window x function can generate a peripheral event or not

| Value | Description |
|---|---|
| 0 | Window x Event is disabled. |
| 1 | Window x 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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.

### 37.7.5. Interrupt Enable Clear

**Name:** INTENCLR
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** RW

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).

**Table 37-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | WIN0 |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | COMP1 | COMP0 |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bit 8 – WINx** Window x Interrupt Enable
Reading this bit returns the state of the Window x interrupt enable. Writing a '0' to this bit has no effect. Writing a '1' to this bit disables the Window x interrupt.

| Value | Description |
|---|---|
| 0 | The Window x interrupt is disabled. |
| 1 | The Window x 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.

### 37.7.6. Interrupt Enable Set

**Name:** INTENSET
**Offset:** 0x14
**Reset:** 0x00000000
**Property:** RW

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).

**Table 37-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | WIN0 |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | COMP1 | COMP0 |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bit 8 – WINx** Window x Interrupt Enable
Reading this bit returns the state of the Window x interrupt enable. Writing a '0' to this bit has no effect. Writing a '1' to this bit enables the Window x interrupt

| Value | Description |
|---|---|
| 0 | The Window x interrupt is disabled. |
| 1 | The Window x 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 enables 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.

### 37.7.7. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** RW

**Note:** Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

**Table 37-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | WIN0 |
| Access | | | | | | | | RW |
| Reset | | | | | | | | 0 |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | COMP1 | COMP0 |
| Access | | | | | | | RW | RW |
| Reset | | | | | | | 0 | 0 |

**Bit 8 – WINx** Window x
This flag is set according to the Window x Interrupt Selection bit group in the WINCTRL register (WINCTRL.WINTSELx) and will generate an interrupt if INTENCLR/SET.WINx is also one.

| Value | Description |
|---|---|
| 0 | No effect |
| 1 | Clears the Window x interrupt flag |

**Bits 0, 1 – COMPx** Comparator x
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.

| Value | Description |
|---|---|
| 0 | No effect |
| 1 | Clears the Comparator x interrupt flag |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.8. Status A

**Name:** STATUSA
**Offset:** 0x1C
**Reset:** 0x00000000
**Property:** R

**Table 37-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | WSTATE0[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 |
|---|---|---|---|---|---|---|---|---|
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | STATE1 | STATE0 |
| Access | | | | | | | R | R |
| Reset | | | | | | | 0 | 0 |

**Bits 16:23 – WSTATEx[7:0]** Window x Current State

| Value | Name | Description |
|---|---|---|
| 0 | ABOVE | Signal is above window |
| 1 | INSIDE | Signal is inside window |
| 2 | BELOW | Signal is below window |

**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 COMPx is enabled and once STATUSB.READYx become one.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.9. Status B

**Name:** STATUSB
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** R

**Table 37-11.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | READY1 | READY0 |
| Access | | | | | | | RW | RW |
| 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.
**Note:**
In single shot mode, the READYx bit is also cleared when a comparison is triggered by software (CTRLB.STARTx = 1). When the comparison is triggered by an event, the READYx bit is set after the first comparison and stay set as long as the comparator is enabled.

| Value | Description |
|---|---|
| 0 | Comparator x output is not ready |
| 1 | Comparator x output is ready. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.10. Debug Control

**Name:** DBGCTRL
**Offset:** 0x24
**Reset:** 0x00000000
**Property:** RW

**Table 37-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | DBGRUN |
| Access | | | | | | | | RW |
| 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.

### 37.7.11. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x28
**Reset:** 0x00000000
**Property:** R

**Table 37-13.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WINCTRL | | |
| Access | | | | | | R | | |
| Reset | | | | | | 0 | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | COMPCTRL | COMPCTRL | ENABLE | SWRST |
| Access | | | | | RW | RW | R | R |
| Reset | | | | | 0 | 0 | 0 | 0 |

**Bit 10 – WINCTRL** WINCTRL x 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.

**Bits 2, 3 – COMPCTRL** COMPCTRL x Synchronization Busy

| Value | Description |
|---|---|
| 0 | Comparison will not start on any incoming event. |
| 1 | Comparison will start on any incoming event. |

**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.

### 37.7.12. Pair n Comparator Control 0

**Name:** COMPCTRL
**Offset:** 0x30 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** RW

**Table 37-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SUT[5:0] | | | | | | OUT[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 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | FLEN[2:0] | | | HYST[1:0] | | | SPEED | SWAP |
| 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 |
|---|---|---|---|---|---|---|---|---|
| | | MUXPOS[2:0] | | | | MUXNEG[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 |
|---|---|---|---|---|---|---|---|---|
| | | RUNSTDBY | INTSEL[1:0] | | SINGLE | | ENABLE | |
| Access | | R/W | R/W | R/W | R/W | | R/W | |
| Reset | | 0 | 0 | 0 | 0 | | 0 | |

**Bits 31:26 – SUT[5:0]** Start-up Time

Each time a comparator is enabled, the comparison will be enabled after the startup time specified by these bits using the formula:
SUT = (20us / ((2^PRESCALER) x t(GCLK_AC))).
Refer to electrical specifications for minimum analog comparator start-up time required to initialize COMPCTRLn.SUT bits to.

**Notes:**

1. If COMPCTRLn.SINGLE=1 even if comparator COMPCTRLn.ENABLE=1 the comparator is turned off to conserve power. When CTRLB.STARTn is set by the user to start the comparison the comparator is enabled, after COMPCTRLn.SUT time has expired the comparison is made and then the comparator is again shut off to conserve power ready for the next time.
2. These bits can be written only while COMPCTRLn.ENABLE is zero.
3. Zero value is not allowed and can lead to unpredictable behavior.

**Bits 25:24 – 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:** If the asynchronous path is selected, the filter settings are ignored.

**Note:** If OUT = 0x2 = Sync, Filtering must be enabled, FLEN>0.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 0 | OFF | The output of COMPn is not routed to the COMPn output |
| 1 | ASYNC | The asynchronous output of COMPn is routed to the COMPn output |
| 2 | SYNC | The synchronous output (including filtering) of COMPn is routed to the COMPn output |

**Bits 23:21 – FLEN[2:0]** Filter Length

These bits configure the filtering for comparator n. COMPCTRLn.FLEN can only be written while COMPCTRLn.ENABLE is zero.

**Note:** If COMPCTRLn.OUT=0x1 (Asyncronous Mode), these bits are ignored.

| Value | Name | Description |
|---|---|---|
| 0 | OFF | No filtering |
| 1 | MAJ3 | 3-bit majority function (2 of 3) |
| 2 | MAJ5 | 5-bit majority function (3 of 5) |

**Bits 20:19 – 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.

| Value | Name | Description |
|---|---|---|
| 0 | HYST10 | 10mV |
| 1 | HYST20 | 20mV |
| 2 | HYST40 | 40mV |
| 3 | HYST60 | 60mV |

**Bit 17 – SPEED** Speed Selection

This bit indicates the speed/propagation delay mode of comparator n. COMPCTRLn.SPEED can be written only while COMPCTRLn.ENABLE is zero.

**Note:** High speed equates to higher operating current as well as faster response time.

| Value | Name | Description |
|---|---|---|
| 0 | HIGH | High speed, high power |
| 1 | LOW | Low speed, low power |

**Bit 16 – 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.

| 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.

| Value | Name | Description |
|---|---|---|
| 0 | PIN0 | I/O pin 0 |
| 1 | PIN1 | I/O pin 1 |
| 2 | PIN2 | I/O pin 2 |
| 3 | PIN3 | I/O pin 3 |
| 4 | INT0 | Internal connection 0 |
| 5 | INT1 | Internal connection 1 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| Value | Name | Description |
|---|---|---|
| 6 | INT2 | Internal connection 2 |
| 7 | INTDAC | Internal DAC |

**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.

| Value | Name | Description |
|---|---|---|
| 0 | PIN0 | I/O pin 0 |
| 1 | PIN1 | I/O pin 1 |
| 2 | PIN2 | I/O pin 2 |
| 3 | PIN3 | I/O pin 3 |
| 4 | INT0 | Internal connection 0 |
| 5 | BANDGAP | Bangap |
| 6 | GND | Ground |
| 7 | INTDAC | Internal DAC |

**Bit 6 – RUNSTDBY** Run in Standby

This bit controls the behavior of the comparator during standby sleep mode. This bit can only be written while COMPCTRLn.ENABLE is zero.

| Value | Description |
|---|---|
| 0 | The comparator n is disabled during sleep. |
| 1 | The comparator n continues to operate during sleep. |

**Bits 5:4 – INTSEL[1:0]** Interrupt Selection

These bits select the condition for comparator n (n=0,1) to generate an interrupt or event. COMPCTRLn.INTSEL can be written only while COMPCTRLn.ENABLE is zero.

| Value | Name | Description |
|---|---|---|
| 0 | TOGGLE | Interrupt on comparator output toggle |
| 1 | RISING | Interrupt on comparator output rising |
| 2 | FALLING | Interrupt on comparator output falling |
| 3 | EOC | Interrupt on end of comparison (single-shot mode only) |

**Bit 3 – SINGLE** Single-Shot Mode

This bit determines the operation of comparator n. COMPCTRLn.SINGLE can be written only while COMPCTRLn.ENABLE is zero.

**Note:** To initiate a single-shot comparison the user's software must write the respective CTRLB.STARTn = 1. Be aware that If COMPCTRLn.SINGLE=1 even if comparator COMPCTRLn.ENABLE=1 the comparator is turned off to conserve power.

When CTRLB.STARTn is set by the user to start the comparison the comparator is enabled, after COMPCTRLn.SUT time has expired the comparison is made and then the comparator is again shut off to conserve power ready for the next time.

| Value | Description |
|---|---|
| 0 | Comparator n operates in continuous measurement mode. |
| 1 | Comparator n operates in single-shot mode. |

**Bit 1 – ENABLE** Enable

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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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.

| Value | Description |
|---|---|
| 0 | Writing a zero to this bit disables comparator n. |
| 1 | Writing a one to this bit enables comparator n. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.13. Dac n Control

**Name:** DACCTRL
**Offset:** 0x38 + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** RW

**Note:** This register can only be written while COMPCTRLn.ENABLE =0.

**Table 37-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | SHEN1 | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | VALUE1[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 |
|---|---|---|---|---|---|---|---|---|
| | SHEN0 | | | | | | | |
| Access | R/W | | | | | | | |
| Reset | 0 | | | | | | | |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | VALUE0[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 31 – SHEN1** DAC1 Sample and Hold Enable Operating Mode
This bit enables the odd DAC low-power operation.
**Note:** If DACCTRL0.SHEN0=1 user must initialize AC.CTRLC.PER and AC.CTRLC.WIDTH accordingly.

**Note:** This bit can be written only while COMPCTRL1.ENABLE is zero.

| Value | Description |
|---|---|
| 0 | Continuous operation mode is enabled. |
| 1 | Sample-and-hold operation mode is enabled. |

**Bits 22:16 – VALUE1[6:0]** DAC1 Output Value
These bits define the scaling factor for odd DAC channel voltage reference. The output voltage, VOUT, is:

$$V_{OUT} = \frac{AVDD \bullet (VALUE)}{128}$$

**Note:** This bit can be written only while COMPCTRL1.ENABLE is zero.

**Bit 15 – SHEN0** DAC0 Sample and Hold Enable Operating Mode
This bit enabled the even DAC low-power operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** If DACCTRLn.SHENn=1 user must initialize AC.CTRLC.PER and AC.CTRLC.WIDTH accordingly.

**Note:** This bit can be written only while COMPCTRL0.ENABLE is zero.

| Value | Description |
|---|---|
| 0 | Continuous operation mode is enabled. |
| 1 | Sample-and-hold operation mode is enabled. |

**Bits 6:0 – VALUE0[6:0]** DAC0 Output Value
These bits define the scaling factor for even DAC channel voltage reference. The output voltage, VOUT, is:

$$V_{OUT} = \frac{AVDD \bullet (VALUE)}{128}$$

These bits can be written only while COMPCTRL0.ENABLE is zero.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 37.7.14. Window Monitor n Control

**Name:** WINCTRL
**Offset:** 0x3C + n*0x04 [n=0..1]
**Reset:** 0x00000000
**Property:** RW

**Table 37-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | WINTSEL[1:0] | | WEN |
| Access | | | | | | RW | RW | RW |
| Reset | | | | | | 0 | 0 | 0 |

**Bits 2:1 – WINTSEL[1:0]** Window n Interrupt Selection
These bits configure the interrupt mode for the comparator n window mode.

| Value | Name | Description |
|---|---|---|
| 0 | ABOVE | Interrupt on signal above window |
| 1 | INSIDE | Interrupt on signal inside window |
| 2 | BELOW | Interrupt on signal below window |
| 3 | OUTSIDE | Interrupt on signal outside window |

**Bit 0 – WEN** Window n Mode Enable
**Note:** If WINCTRL.WIN=1 then both pair of appropriate comparators must have identical COMPCTRLn.SINGLE bit value settings.

| Value | Description |
|---|---|
| 0 | Window mode is disabled for Comparator 0 and 1 |
| 1 | Window mode is enabled for Comparator 0 and 1 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 38. Timer/Counter for Control Applications (TCC)

## 38.1. Overview

The Timer/Counter for Control Applications (TCC) module provides a set of timing and counting related functionality, such as the generation of periodic waveforms, the capturing of a periodic waveform's frequency/duty cycle, software timekeeping for periodic operations, waveform extension control, fault detection, and so on. The counter size of the TCC modules can be 16-bit or 32-bit depending on the TCC instance. Refer to the *Configuration Summary* for details on TCC instances.

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 or 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 (PWM).

TCC contains waveform extension features for motor control, ballast control, LED control, H-bridge control, power converters, and other types of power control applications. These extensions allow for low-side and high-side outputs 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, deactivating output signals from the microcontroller. These signals can be used to shutdown external drivers connected to the microcontroller.

**Note:** The number of TCC instances, TCC channel numbers, and TCC extensions are device specific. The TCC configurations, such as channel numbers and features, may be reduced for some of the TCC instances.

References:

- Device Configuration Summary

## 38.2. Features

The following are key features of the module:

- Up to 2 Compare/Capture Channels (CC) with:
  - Double buffered period setting
  - Double buffered compare or capture channel
  - Circular buffer on period and compare channel registers
- Input Capture:
  - Event capture
  - Frequency capture
  - Pulse-width capture
- Waveform Generation:
  - Frequency generation
  - Single-slope Pulse-Width Modulation (PWM)
  - Dual-slope PWM with half-cycle reload capability
- Waveform Extensions:
- Fault Protection for Safe Disabling of Drivers:
  - Two recoverable fault sources
  - Two non-recoverable fault sources
  - Debugger can be a source of non-recoverable fault


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Input Events:
  - Two input events (TCCx_EV_0 , TCCx_EV_1) for counter
  - One input event (TCCx_MC_y) for each channel (y = 0,1,2...7)
- Output Events:
  - Three output events (Count, re-trigger, and overflow) are available for counter
  - One compare match or input capture event output for each channel
- Interrupts:
  - Overflow and re-trigger interrupt
  - Compare match/input capture interrupt
  - Interrupt on fault detection
- Can be used with DMA and can Trigger DMA Transactions

### 38.2.1. Feature Summary

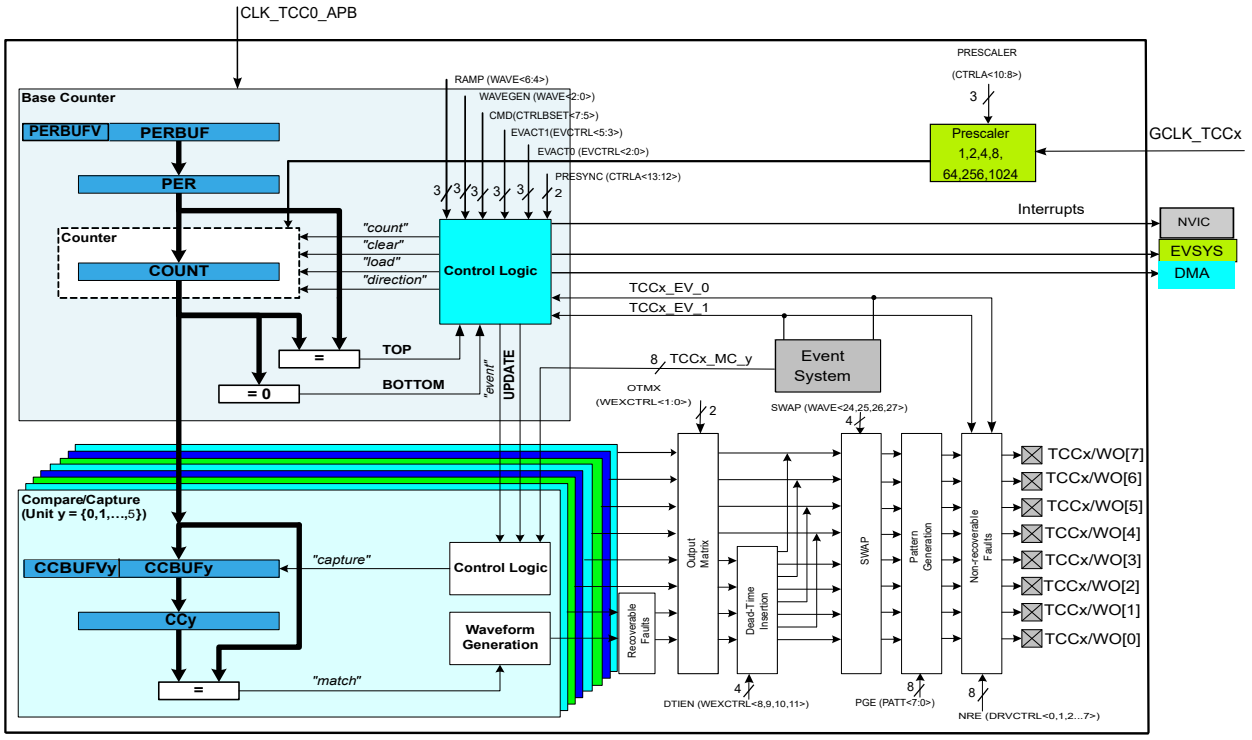
| TCC Instance | Capture-Compare Channels | Waveform Output Channel Pins | Output Channels that Support Dithering | (OTMX) Output Matrix Channels Available | (DTI) Output Channels that Support Dead Time Insertion | (SWAP) Outputs Channels that Support HS / LS swapping | (PG) Output Channels that Support Pattern Generation | Counter Size |
|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 16-bit |
| 1 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 16-bit |
| 2 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 16-bit |
| 3 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 16-bit |
| 4 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 16-bit |
| 5 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 16-bit |
| 6 | 2 | 2 | 0 | 0 | 0 | 0 | 0 | 16-bit |

## 38.3. Block Diagram

The following block diagram illustrates the Timer/Counter for Control Applications:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-1.** Timer/Counter for Control Applications Block Diagram




## 38.4. Signal Description

| PIN NAME (1,2) | SIGNAL TYPE | Description |
|---|---|---|
| TCC0_WO0 | Digital I/O | TCC0 channel 0, waveform output |
| TCC0_WO1 | Digital I/O | TCC0 channel 1, waveform output |
| TCC1_WO0 | Digital I/O | TCC1 channel 0, waveform output |
| TCC1_WO1 | Digital I/O | TCC1 channel 1, waveform output |
| TCC2_WO0 | Digital I/O | TCC2 channel 0, waveform output |
| TCC2_WO1 | Digital I/O | TCC2 channel 1, waveform output |
| TCC3_WO0 | Digital I/O | TCC3 channel 0, waveform output |
| TCC3_WO1 | Digital I/O | TCC3 channel 1, waveform output |
| TCC4_WO0 | Digital I/O | TCC4 channel 0, waveform output |
| TCC4_WO1 | Digital I/O | TCC4 channel 1, waveform output |
| TCC5_WO0 | Digital I/O | TCC5 channel 0, waveform output |
| TCC5_WO1 | Digital I/O | TCC5 channel 1, waveform output |
| TCC6_WO0 | Digital I/O | TCC6 channel 0, waveform output |
| TCC6_WO1 | Digital I/O | TCC6 channel 1, waveform output |

**Note:**

1. Refer to the *Pinout* chapter for additional information on the pin mapping for this peripheral. One signal can be mapped on several pins.

## 38.5. Functional Description

### 38.5.1. Principle of Operation

The following definitions are used throughout the documentation:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-1.** 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. (i.e. 0x00000000) |
| MAX | The counter reaches maximum when it contains all ones. (i.e. 0xFFFFFFFF) |
| 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." |

There are up to 2 compare/capture (CC) channels starting from CC0 to CCn. The number of CC channels can be different on different instances of a TCC.

The Counter register (COUNT), Period registers with Buffer (PER and PERBUF), and Compare and Capture registers with buffers (CCy and CCBUFy) are 32-bit registers, on each TCC instance( if the TCC instance is a 16-bit timer/counter, the excess bits read as zero). Each Buffer register has a corresponding Buffer Valid flag in the STATUS register 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 interrupt requests, request DMA transactions, or generate events for the Event System. In Waveform Generator operation, 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 Recoverable Fault event inputs are connected to the MC0 and MC1 event lines. 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 for different types of motors, ballasts, LEDs, H-bridges, power converters, and other types of power switching applications, the following independent units are implemented in some of the TCC instances:

- Recoverable faults and non-recoverable faults

See also TCC Block Diagram.

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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

application. This is typically used for instant and predictable shut down and disabling high current or voltage drives. A non-recoverable fault can be recovered only by software.

The count event sources TCCx_EV_0 connected to TCE0 input and TCCx_EV_1 connected to TCE1 input 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*.

References:

- EVSYS

### 38.5.2. Basic Operation

#### 38.5.2.1. 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:

- The Software Reset and Enable bits in the Control A register (CTRLA.SWRST and CTRLA.ENABLE)

The following registers are synchronized when written:

- The Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- The Status register (STATUS)
- The Waveform register (WAVE)
- The Count Value register (COUNT)
- The Period Value and Period Buffer Value registers (PER and PERBUF)
- The Compare/Capture Channel y and Compare/Capture Channel y Buffer Value registers (CCy and CCBUFy)

The following registers are synchronized when read:

- The Control B Clear and Control B Set registers (CTRLBCLR and CTRLBSET)
- The Count Value register (COUNT): synchronization is done on demand through READSYNC command (CTRLBSET.CMD)
- The Waveform register (WAVE)
- The Period Value and Period Buffer Value registers (PER and PERBUF)
- The Compare/Capture Channel y and Compare/Capture Channel y Buffer Value registers (CCy and CCBUFy)

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.

Reference:

- Register Synchronization

#### 38.5.2.2. Initialization

The following registers are enable-protected, meaning that they can only be written when TCC is disabled (CTRLA.ENABLE = 0):

- The Recoverable Fault Control registers (FCTRLA and FCTRLB)
- The Waveform Extension Control register (WEXCTRL)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- The Drive Control register (DRVCTRL)
- The Event Control register (EVCTRL)

Register Enable-protection is denoted by the *Enable-Protected* property in the register description.

The following register bits are enable-protected, meaning that they can only be written when the TCC is disabled (CTRLA.ENABLE = 0):

- The Capture Channel x Enable bits in the Control A register (CTRLA.CPTEN[31:24])
- DMA One-shot Trigger Mode (CTRLA.DMAOS)
- Full Cycle (CTRLA.FCYCLE)
- Auto Lock (CTRLA.ALOCK)
- Prescaler and Counter Synchronization Selection (CTRLA.PRESCSYNC)
- Run in Standby (CTRLA.RUNSTDBY)
- Prescaler (CTRLA.PRESCALER)
- Enhanced Resolution (CTRLA.RESOLUTION)

When the CTRLA.ENABLE bit (CTRLA <1>) is '0', the Enable-Protected bits in the CTRLA register can be written at the same time the CTRLA.ENABLE bit (CTRLA <1>) is written to '1'. However, when the CTRLA.ENABLE bit (CTRLA <1>) is '1' these bits can not be written at the same time CTRLA.ENABLE bit (CTRLA <1>) is written to '0'. the enable-protection is denoted by the *Enable-Protected* property in the register description.

Before TCC is enabled, it must be configured as outlined by the following steps:

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a ‘1’ to the TCCx_bit in the Peripheral Bus Clock Enable Maskn Register of the MCLK module.
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 (i.e., for the Channel 0 CTRLA.CPTEN0 bit (CTRLA <24>), for the Channel 1 CTRLA.CPTEN1 bit (CTRLA <25>) and so on).

Optionally, the following configurations can be set before enabling TCC:

1. Select PRESCALER setting in the Control A register (CTRLA.PRESCALER bits (CTRLA <10:8>)).
2. Select Prescaler Synchronization setting in Control A register (CTRLA.PRESYNC bits (CTRLA <13:12>)).
3. If down-counting operation is desired, write the Counter Direction bit in the Control B Set register (CTRLBSET.DIR bit (CTRLBSET <0>)) to '1'. In this case, the COUNT register must be initialized with the desired TOP value.
4. Select the Waveform Generation operation in the WAVE register (WAVEGEN bits (WAVE <2:0>)).
5. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y = 0,1,2,..7. For example, when the CC0 register is used for the duty cycle, the corresponding WAVE.POL0 bit (WAVE<16>) decides the polarity.
6. The waveform output for each individual channel can be inverted by configuring the corresponding Waveform Output Invert Enable DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when CC0 register is used for the duty cycle, corresponding DRVCTRL.INVEN0 bit (DRVCTRL <16>) decides inversion.

**Note:** See different PWM modes for detailed information.

#### 38.5.2.3. 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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 register for details.

The TCC should be disabled before the TCC is reset to avoid undefined behavior.

### 38.5.2.4. 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_TCC_COUNT.

### 38.5.2.5. Counter Operation

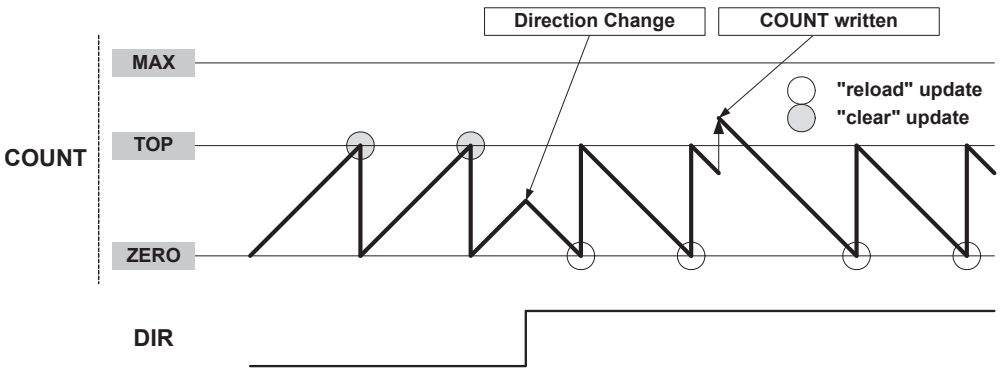
Depending on the mode of operation, the counter is cleared, reloaded, incremented, or decremented at each TCC clock input (CLK_TCC_COUNT). A counter clear or reload mark the end of current counter cycle and the start of a new one.

The counting direction is set by the Direction bit in the Control B Set/Clear registers (CTRLBSET and CTRLBCLR). If the bit reads zero, it is counting up and if the bit reads one, it is counting down.

The counter will count up or 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 or TOP 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 bit is set. When TC is counting down, the COUNT register must be initialized to TOP value (PER or CC0 value depending on the mode).

INTFLAG.OVF bit 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). The One-Shot feature is explained in the section “Additional Features”.

**Figure 38-2.** Counter Operation




It is possible to change the counter value (by writing directly in the COUNT register) even when the counter is running.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When starting the TCC, the COUNT value will always be ZERO or TOP, depending on direction set by CTRLBSET.DIR bit (CTRLBSET <0> or CTRLBCLR.DIR bit (CTRLBCLR <0>), unless a different value has been written to it, or the TCC has been stopped at a value other than ZERO or TOP.

The write access has higher priority than count, clear, or reload. The direction of the counter can also be changed during normal operation. See the previous figure.

**Stop Command**

A stop command can be issued from software by using TCC Command bits in Control B Set register (CMD bits (CTRLBSET <7:5>)=0x2, STOP).

When a stop is detected while the counter is running, and if CTRLA.FCYCLE bit (CTRLA <16>) = 0 the counter stops immediately maintaining its current value, however, if CTRLA.FCYCLE bit (CTRLA <16>)=1, it waits until end of current cycle to stop on a start value.

If the waveform generation operation (WG) is used, all waveforms are set to a state defined in Non-Recoverable State y Output Enable bit and Non- Recoverable State y Output Value bit in the Driver Control register (DRVCTRL.NREy and DRVCTRL.NRVy), and the Stop bit in the Status register is set (STATUS.STOP bit (STATUS <0>)).

**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.EVACT0 bits (EVCTRL <2:0>)=0x3, STOP).

When a pause is detected, the counter will maintain its current value and all waveforms keep their current state when (CTRLA.FCYCLE (CTRLA <16>)=0), or wait to the end of the current cycle to stop on a start value and all waveforms keep their initial state when (CTRLA.FCYCLE (CTRLA <16>)= 1), until a start event action is detected: Input Event Action 0 bits in Event Control register (EVCTRL.EVACT0 bits (EVCTRL <2:0>) = 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 (CTRLBSET <7:5>)=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.EVACT0 bits (EVCTRL <2:0>)=0x1, RETRIGGER and EVCTRL.EVACT1 bits (EVCTRL <5:3>) =0x1, RETRIGGER).

When the command is detected during counting operation, the counter will be reloaded or cleared, depending on the counting direction (CTRLBSET.DIR (CTRLBSET <0>) or (CTRLBCLR.DIR (CTRLBCLR <0>)). The Re-Trigger bit in the Interrupt Flag Status and Clear register will be set (TRG bit (INTFLAG <1>)). 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 bit (EVCTRL <9>)). 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.EVACT0 bits (EVCTRL <2:0>)=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 (EVCTRL <2:0>)=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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:**
When a start event action is configured in the Event Action bits in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=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 (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

The count event action is selected by the Event Action 0 bit group in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=0x5, COUNT).

**Direction Event Action**

The direction event action can be selected in the Event Control register (EVCTRL.EVACT1 (EVCTRL <5:3>)=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 (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)) and the direction bit value is updated accordingly.

**Increment Event Action**

The increment event action can be selected in the Event Control register (EVCTRL.EVACT0 (EVCTRL <2:0>)=0x4, INC) and can change the Counter state when an event is received. When the TCCx_EV_0 (TCE0) event is received, the counter increments, irrespective of direction setting (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

**Decrement Event Action**

The decrement event action can be selected in the Event Control register (EVCTRL.EVACT1 (EVCTRL <5:3>)=0x4, DEC) and can change the Counter state when an event is received. When the TCCx_EV_1 (TCE1) event is received, the counter decrements, irrespective of direction setting (CTRLBSET.DIR (CTRLBSET <0>) or CTRLBCLR.DIR (CTRLBCLR <0>)).

**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.NREy and DRVCTRL.NRVy). TCCx_EV_0 input event on TCE0 and TCCx_EV_1 input events on 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.

### 38.5.2.6. 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 (CCy) for compare operations, the counter value is continuously compared to the values in the CCy registers. This can be used for timer or for waveform operation.

The Channel y Compare/Capture Buffer Value (CCBUFy) registers provide double buffer capability. The double buffering synchronizes the update of the CCy register with the buffer value at the UPDATE condition or a force update command (CTRLBSET.CMD (CTRLBSET <5:3>)=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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 38.5.2.6.1. 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 Operation in the Waveform Generation Operation bits in Waveform register (WAVE.WAVEGEN (WAVE <2:0>)).
2. Optionally invert the waveform output WO[y] by writing the corresponding Waveform Output y Inversion bit in the Driver Control register (DRVCTRL.INVENy).
3. Configure the pins with the I/O Pin Controller. Refer to *PORT - I/O Pin Controller* chapter 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 CCy value. On a comparison match, the Match or Capture Channel y bit in the Interrupt Flag Status and Clear register (INTFLAG.MCy) will be set on the next zero-to-one transition of GCC_TCCx (see Normal Frequency Operation). An interrupt and/or event can be generated on the same condition if Match/Capture occurs, i.e. INTENSET.MCy and/or EVCTRL.MCEOy is '1'. Both interrupt and event can be generated simultaneously. The same condition generates a DMA request.

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 Operations, the update time occurs on counter wraparound, on overflow, on underflow, or on re-trigger.

The table below shows the update counter and overflow event/interrupt generation conditions in different Waveform Generation Operations.

**Table 38-2.** Counter Update and Overflow Event/interrupt Conditions

| Name | Operation | TOP | Update | Output Waveform | | OVF Interrupt Flag/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 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-2.** Counter Update and Overflow Event/interrupt Conditions (continued)

| Name | Operation | TOP | Update | Output Waveform | | OVF Interrupt Flag/Event | |
|---|---|---|---|---|---|---|---|
| | | | | On Match | On Update | Up | Down |
| 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.

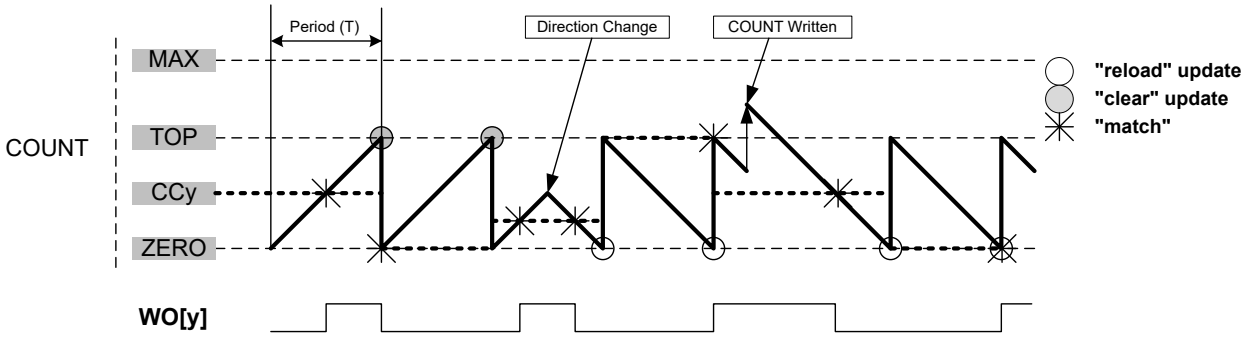
References:

- Circular Buffer
- PORT I/O

### 38.5.2.6.2. Normal Frequency (NFRQ)

For Normal Frequency generation, the period time (T) is controlled by the period register (PER). The waveform generation output (WO[y]) is toggled on each compare match between COUNT and CCy, and the corresponding Match or Capture Channel y Interrupt Flag (INTFLAG.MCy) will be set.

**Figure 38-3.** Normal Frequency Operation




The following steps must be performed to operate the TCC in Normal Frequency (NFRQ) Waveform Generation Operations.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired Prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Select matrix routing to desired port pins for generated output waveform, by configuring OTMX bits (WEXCTRL.WEXCTRL<1:0>).
5. Set Waveform Generation Operations to Normal Frequency Operation (NRFQ) by clearing WAVE.WAVEGEN bits (WAVE<2:0> =0)

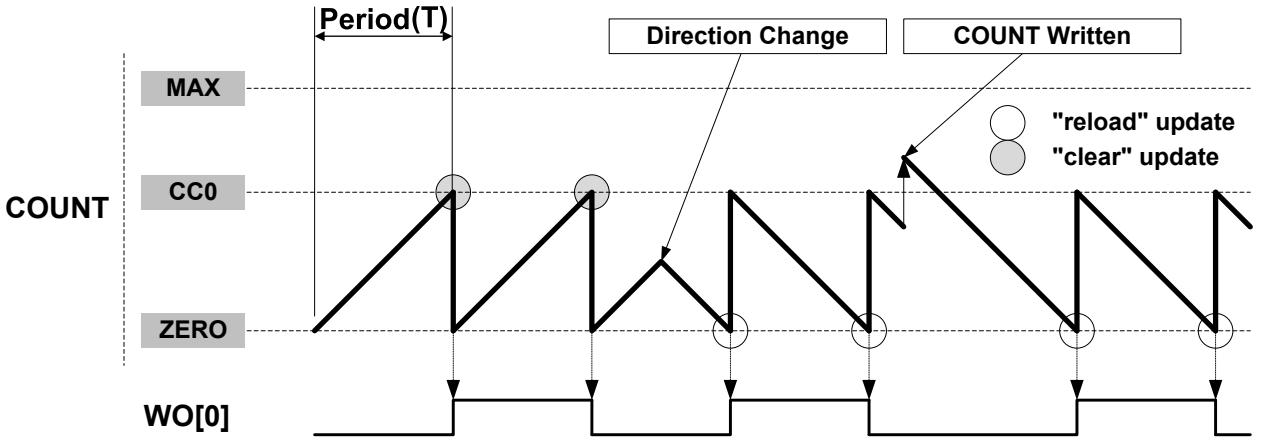

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

6. Load the selected Compare/Capture (CCy) register (e.g. CC0<31:0>) with the desired compare match value. The generated output will toggle on this match.
7. Load the period register PER<31:0> with the desired time period value.
8. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> =1) The waveform output of a channel CCy can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
9. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling TCCx IRQ.
10. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
11. To know how to clear interrupts, see Interrupts.

#### 38.5.2.6.3. 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 38-4.** Match Frequency Operation




The following steps must be performed to operate the TCC in Match Frequency (MFRQ) Operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting CHEN bit PCHCTRL31.PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set Waveform Generation Operation to MATCH Frequency Operation (MFRQ) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 0x1).
5. Load the selected Compare/Capture (CC) register (e.g. CC0<31:0>) with the desired compare match value. The generated output will toggle on this match, PER register is not used in this operations.
6. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> =1).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

7. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
8. The waveform output can be inverted by configuring DRVCTRL.INVEN0 bit (DRVCTRL<16>).
9. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
10. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
11. To know how to clear interrupts, see Interrupts.

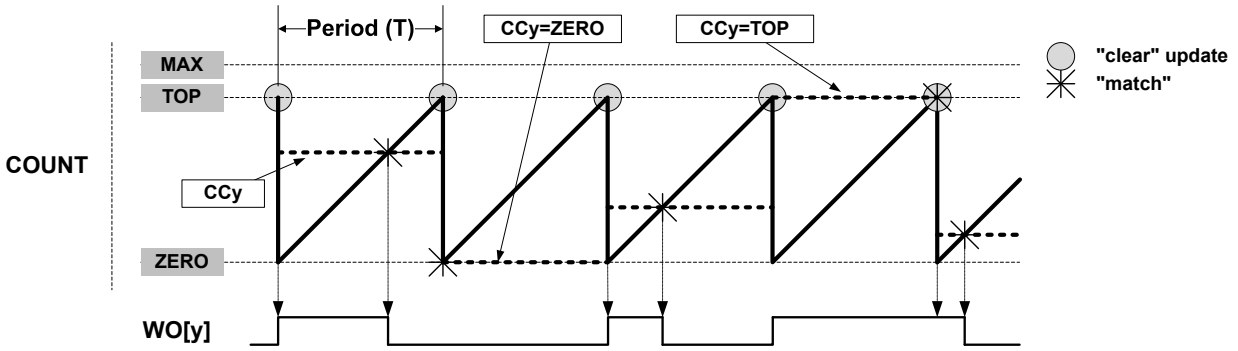
#### 38.5.2.6.4. Normal Pulse-Width Modulation (NPWM)

NPWM uses single-slope PWM generation.

#### 38.5.2.6.5. Normal Pulse-Width Modulation (NPWM) Single-Slope Operation

For single-slope PWM generation, the period setting (TOP) is controlled by the PER register, and CCy controls the duty cycle of the generated waveform output. When up-counting, the WO[y] is set at start or at compare match between the COUNT and TOP values, and cleared on compare match between COUNT and CCy register values. When down-counting, the WO[y] is cleared at start or at compare match between the COUNT and ZERO values, and set on compare match between COUNT and CCy register values.

**Figure 38-5.** Single-Slope PWM Operation




The following equation calculates the exact resolution for a single-slope PWM ($R_{PWM_SS}$) waveform:

$$R_{PWM_SS} = \frac{\log(PER+1)}{\log(2)}$$

The PWM frequency depends on the Period register value (PER) and the peripheral clock frequency ($f_{GCLK_TCCx\ =\ 1/GCLK_TCCx}$), and can be calculated by the following equation:

$$f_{PWM_SS} = \frac{f_{GCLK_TCCx}}{N(PER+1)}$$

Where N represents the prescaler divider used CTRLA.PRESCALER (CTRLA<10:8>). The selected prescaler value can be 1, 2, 4, 8, 16, 64, 256, or 1024.

The following steps must be performed to operate the TCC in single-slope PWM generation operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set Waveform Generation Operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Load the selected Compare/Capture (CCy) register (e.g. CC0<31:0>) with the desired PWM duty cycle value.
6. Load the period register PER<31:0> with the desired time period value.
7. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1). To change the counter direction down set counter direction bit CTRLBSET.DIR(CTRLBSET<0> =1).
8. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
9. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y= 0,1,2,..7. For example, when the CC0 register is used for the duty cycle, the corresponding POL0 bit (WAVE<16>) decides the polarity.
10. The waveform output for each individual channel can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
11. If overflow interrupt is used, set OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
12. Enable TCC by setting ENABLE bit (CTRLA<1>).
13. To know how to clear interrupts, see Interrupts.

#### 38.5.2.6.6. Normal Pulse-Width Modulation (NPWM) Dual-Slope Operation

For dual-slope PWM generation, the period setting (TOP) is controlled by PER, while CCy 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 depends on DS mode (DSTOP, DSBOTTOM, DSBOTH or DSCRITICAL). Depending on Dual-Slope mode (DSTOP, DSBOTTOM, DSBOTH, or DSCRITICAL), 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.

References:

- Circular Buffer

The following steps must be performed to operate the TCC in Dual-slope PWM generation operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the APB Mask register of the MCLK (i.e. enable CLK_TCC0_APB for TCC0 by setting TCC0_ bit (APBCMASK<3>), CLK_TCC1_APB for TCC1 by setting TCC1_ bit (APBCMASK<4>) etc.).
2. Enable Generic clock for TCC Instance (e.g. enable GCLK_TCC0 for TCC0 by setting PCHCTRL31.CHEN bit PCHCTRL31<6>, GCLK_TCC1 for TCC1 by setting PCHCTRL32.CHEN bit PCHCTRL32<6> etc.).
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Select prescaler Synchronization PRESCSYNC bits (CTRLA <13:12>).
5. Set waveform generation operation to Dual Slope PWM by configuring WAVE.WAVEGEN bits (WAVE<2:0>)
   a. For DSCRITICAL operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 4)

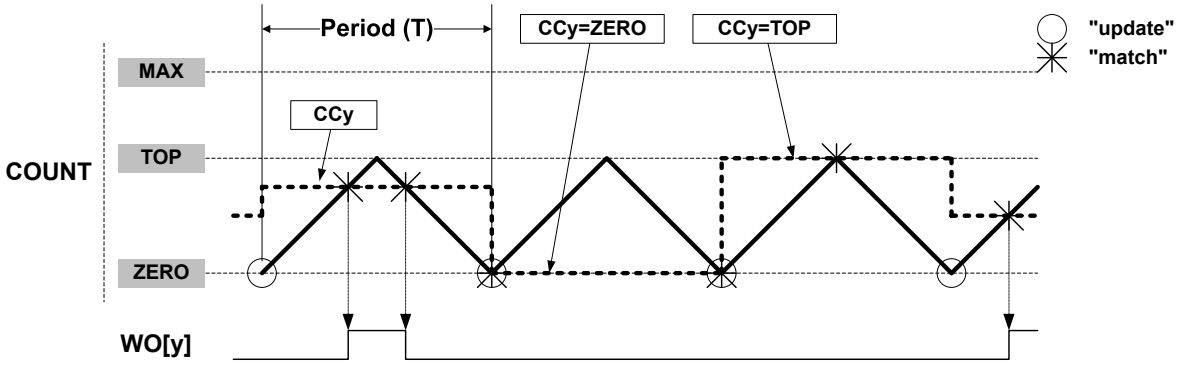

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

b. For DSBOTTOM operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 5)
c. For DSBOTH operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 6)
d. For DSTOP operation, set the value of WAVE.WAVEGEN bit (WAVE<2:0> = 7)

6. Load the selected Compare/Capture (CCy) register (i.e. CC0<31:0>) with the desired PWM duty cycle value.
7. Load the period register PER<31:0> with the desired time period value.
8. Set Counter to count in up direction by clearing DIR bit (CTRLBCLR<0> = 1).
9. If DSBOTH operation (WAVEGEN bit (WAVE<2:0> = 6)) is set, enable Circular buffer by setting CIPEREN bit (WAVE<7>).
10. Select matrix routing to desired port pins for generated output waveform, by configuring WEXCTRL.OTMX bits (WEXCTRL<1:0>).
11. The dead time insertion on output matrix can be enabled by setting WEXCTRL.DTIENy bit, where y=0,1,2,3.
12. Higher side and Lower side PWM Dead-time is programmed in WEXCTRL.DTHS (WEXCTRL<30:24>) and WEXCTRL.DTLS (WEXCTRL<23:16>) respectively.
13. The output polarity for each individual channel can be changed by configuring the corresponding WAVE.POLy bit, where y= 0,1,2,..7. For example, when the CC0 register is used for the duty cycle, the corresponding POL0 bit (WAVE<16>) decides the polarity.
14. The waveform output for each individual channel can be inverted by configuring the corresponding Waveform Output Invert Enable bit DRVCTRL.INVENy bits where y = 0,1,2...7. For example, when the CC0 register is used for the duty cycle, the corresponding INVEN0 bit (DRVCTRL<16>) decides inversion.
15. If overflow interrupt is used, set OVF bit (INTENSET<0>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
16. Enable TCC by setting ENABLE bit (CTRLA<1>).
17. For instructions on how to clear interrupts, see Interrupts.

Dual-slope Critical (DSCRITICAL) PWM operation is explained below.

**Figure 38-6.** 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 prescaler can be selected by CTRLA.PRESCALER bits (CTRLA<10:8>). The selected prescaler value can be 1, 2, 4, 8, 16, 64, 256, 1024.. 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 (CCy) register value and the peripheral clock frequency ($f_{GCLK_TCCx}$), and can be calculated by the following equation:

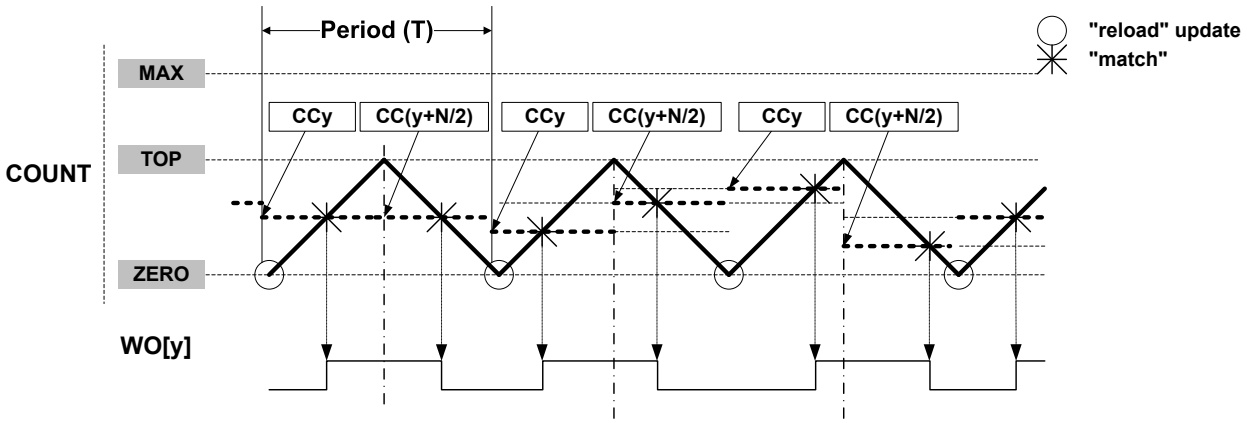
$$P_{PWM_DS} = \frac{2N \cdot (TOP - CCy)}{f_{GCLK_TCCx}}$$

**Note:** In DSTOP, DSBOTTOM and DSBOTH operation, when TOP is lower than MAX/2, the MSB bit of CCy defines the Ramp on which the CCy Match interrupt or event is generated. (Rising if CCy[MSB] = 0, falling if CCy[MSB] = 1.)

### 38.5.2.6.7. Dual-Slope Critical PWM Generation

Critical generation operation allows generation of non-aligned centered pulses. In this operation, the period time is controlled by PER while CCy control the generated waveform output edge during up-counting and CC(y+CC_NUM/2) control the generated waveform output edge during down-counting.

**Figure 38-7.** Dual-Slope Critical Pulse Width Modulation (N=CC_NUM)

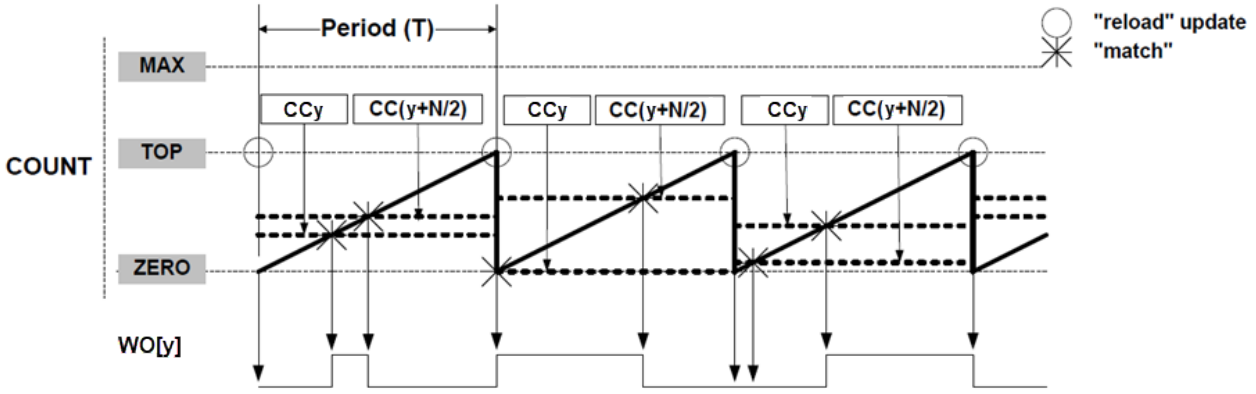



### 38.5.2.6.8. Dual-Compare PWM Generation

Dual compare PWM generation allows generation of pulses unaligned on start or end of a period. In this operation, the period time is controlled by PER, while CCy controls the waveform output leading edge and the CC(y+CC_NUM/2) controls the waveform output trailing edge.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-8.** Dual-Compare Pulse Width Modulation (N=CC_NUM)




### 38.5.2.6.9. Output Polarity

The polarity (WAVE.POLy) 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.

**Table 38-3.** Waveform Generation Set/Clear Conditions

| Waveform Generation Operation | DIR | POL | Waveform Generation Output Update | |
|---|---|---|---|---|
| | | | Set | Clear |
| Single-Slope PWM | 0 | 0 | Timer/counter matches TOP | Timer/counter matches CC |
| | | 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.POLy value represents the initial state of the waveform output.

### 38.5.2.7. Double Buffering

The Period (PER) and Compare Channels (CCy) registers are all double buffered. Each buffer register has a buffer valid ( PERBUFV(STATUS<7>) and CCBUFVy) bit in the STATUS register, which indicates that the Buffer register contains a valid value that can be copied into the corresponding register. When a Buffer Valid Status flag (PERBUFV or CCBUFVy) is set and the corresponding SYNCBUSY bit is set (SYNCBUSY.PATT(SYNCBUSY<5>),SYNCBUSY.PER(SYNCBUSY<7>) or SYNCBUSY.CCy), a write to the respective PER/PERBUF or CCy/CCBUFy registers will generate a PAC error.

When the Buffer Valid Flag bit in the STATUS register is '1' and the Lock Update bit in the CTRLBCLR register is cleared (writing CTRLBCLR.LUPD(CTRLBCLR<1>) to '1'), update of register by its buffer register is allowed: Data from the buffer register will be copied into the corresponding register under hardware UPDATE conditions, then the Buffer Valid flag bit in the STATUS register is automatically cleared by the hardware.

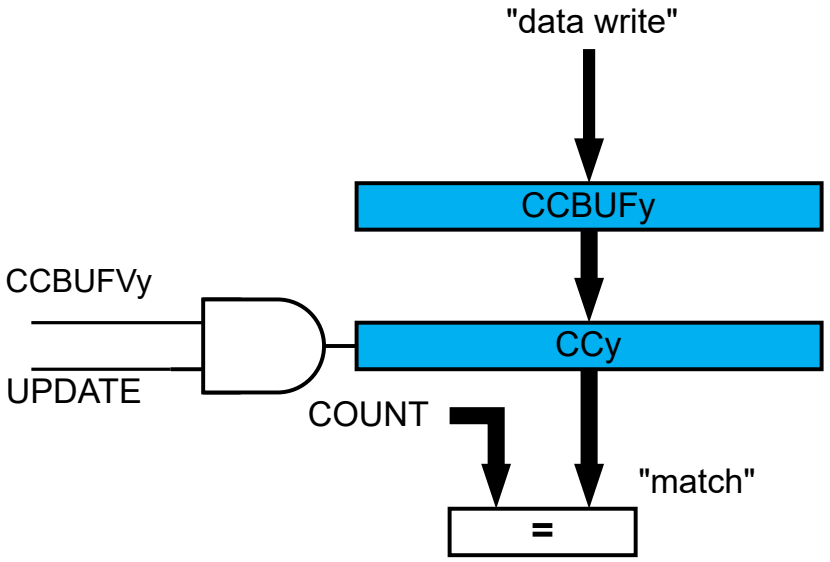
When the Buffer Valid Flag bit in the STATUS register is '1' (that is, the buffer register contains a valid value), and the Lock Update bit in the CTRLB register is set to '1', (writing CTRLBSET.LUPD(CTRLBSET<1>) to '1'), update of a register by its Buffer register is disabled: Data from Buffer register is not copied into the corresponding register on any update conditions and the Buffer Valid Flag bit in the STATUS register stay unchanged.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Note:** Software update command (CTRLBSET.CMD(CTRLBSET<7:5>) = 0x3) act independently of LUPD value.

A compare register is double buffered as shown in the following figure.

**Figure 38-9.** Compare Channel Double Buffering

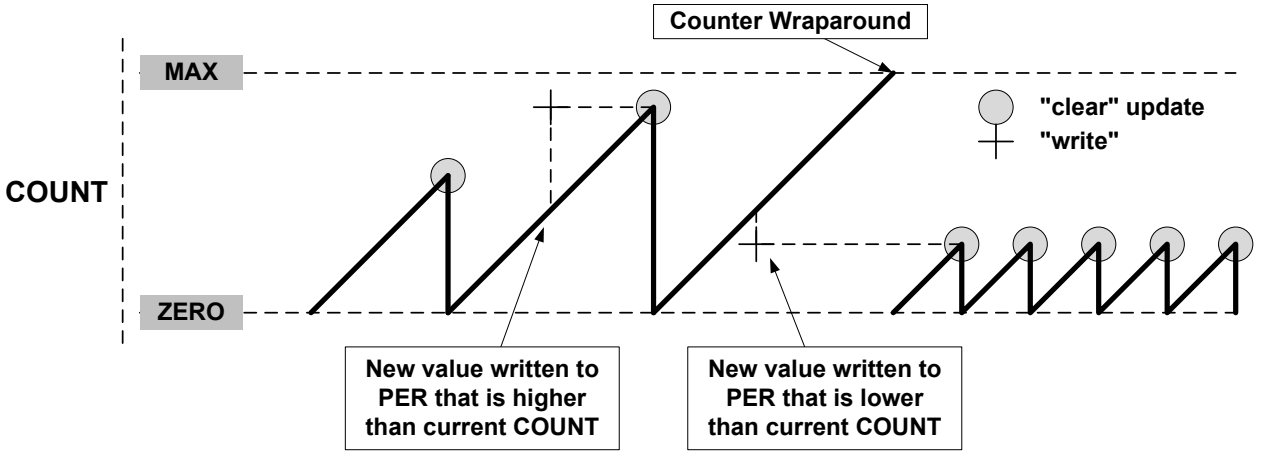



The registers (PER/CCy) and corresponding Buffer registers (PERBUF/CCBUFy) are available in the I/O register map, and the double buffering feature is not mandatory. Double buffering feature can be bypassed by directly writing on the (/PER/CCy) registers.

### Changing the Period

The counter period can be changed by writing a new TOP value to the register that decides period (PER or CCy, depending on the Waveform Generation Operation), however period update on registers (PER or CCy) is effective after the synchronization delay, irrespective of whether double buffer is enabled or disabled.

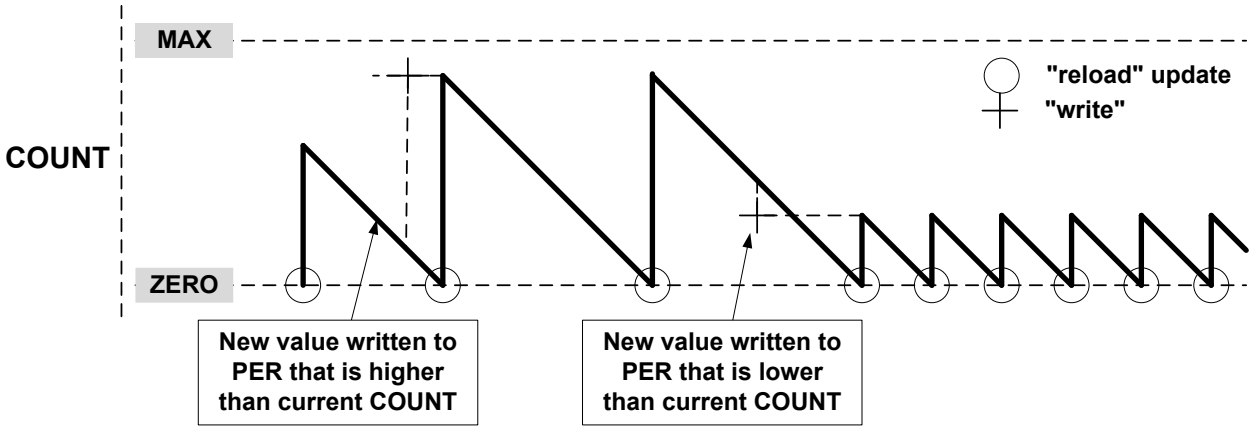
**Figure 38-10.** Unbuffered Single-Slope Up-Counting Operation





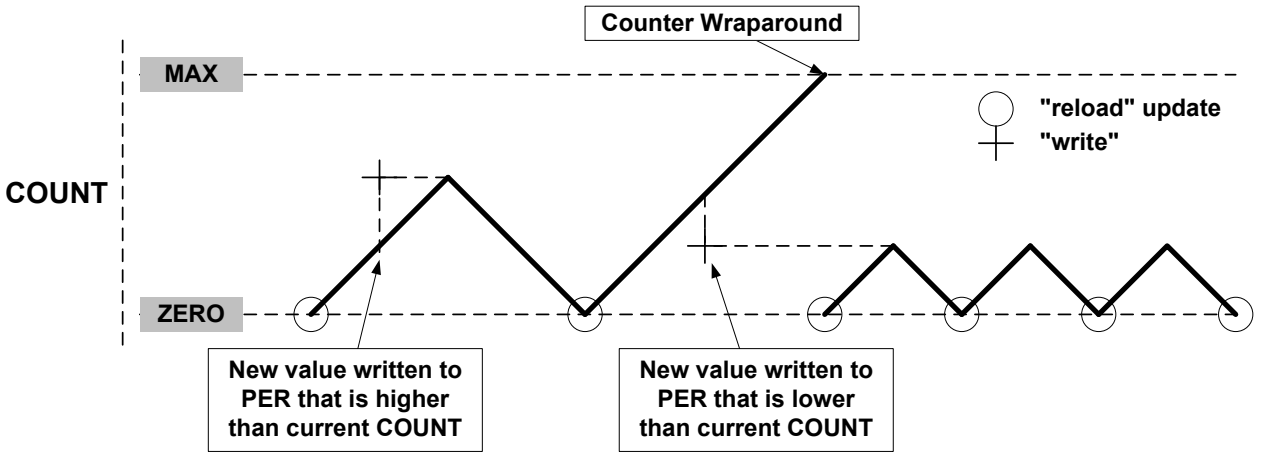
© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-11.** Unbuffered Single-Slope Down-Counting Operation




A counter wraparound can occur in any Waveform Generation Operation when up-counting without buffering, see the previous figure. 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. Similarly, unbuffered down-counting operation is shown in the previous figure. Unbuffered operation in dual-slope mode is shown in the following figure.

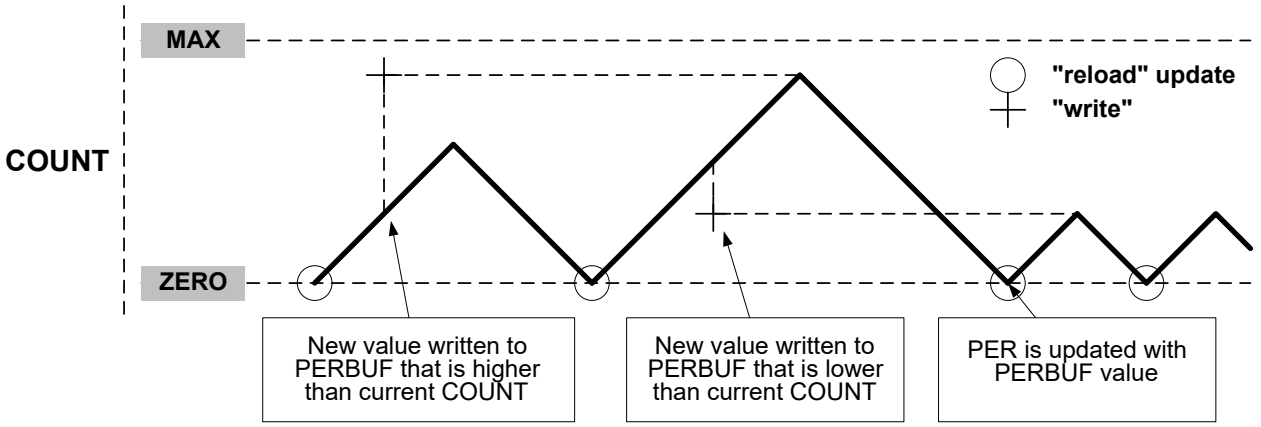
**Figure 38-12.** Unbuffered Dual-Slope Operation




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 *Changing the Period Using Buffer*. This prevents wraparound and the generation of odd waveforms.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-13.** Changing the Period Using Buffering



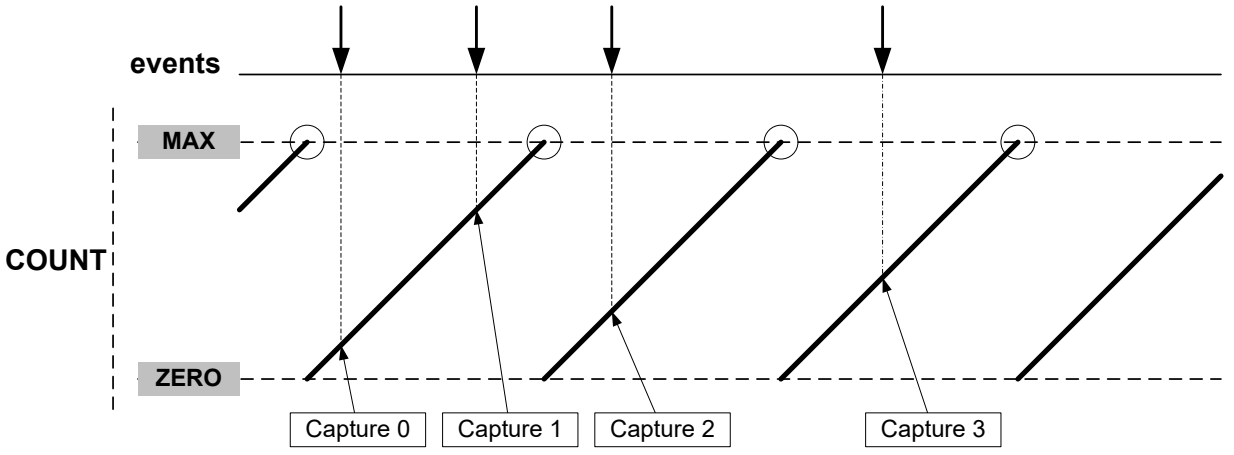

### 38.5.2.8. Capture Operations

To enable and use capture operations, the Match or Capture Channel y Event Input Enable bit (MCEIy) in the Event Control register must be written to '1'. The capture channels to be used must also be enabled in the Capture Channel y Enable bit (CPTENy) in the Control A register (CTRLA.CPTENy) 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.

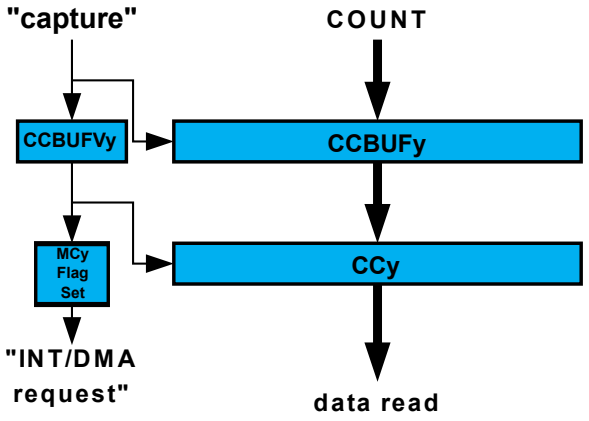
**Figure 38-14.** Input Capture Timing




For input capture, the Buffer register and the corresponding CCy act like a FIFO. When CCy is empty or read, any content in CCBUFy is transferred to CCy. The Buffer Valid flag (STATUS.CCBUFVy) is passed to set the CCy Interrupt flag (INTFLAG.MCy) and generate the optional interrupt, event, or DMA request. The CCBUFy register value cannot be read, all captured data must be read from the CCy register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-15.** Capture Double Buffering



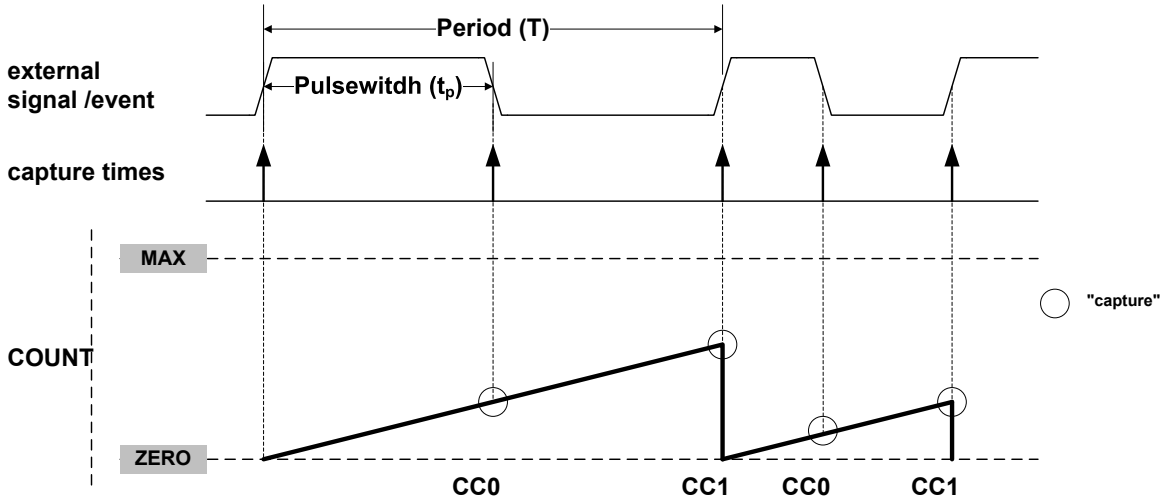

The TCC can detect capture overflow of the input capture channels. When a new capture event is detected while the Capture Buffer Valid flag, CCBUFVy (For example STATUS.CCBUFV0 for CCBUF0 buffer register) is still set, the new timestamp will not be stored and INTFLAG.ERR(INTFLAG<3>) will be set.

### Pulse-Width and Period (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:

$$f = \frac{1}{T} \quad , \quad dutyCycle = \frac{t_p}{T}$$

**Figure 38-16.** PWP Capture




Selecting Pulse Width and Period (PWP) in the Timer/Counter Event Input 1 Action bit group in the Event Control register (EVCTRL.EVACT1(EVCTRL <5:3>)) enables the TCC to perform one capture action on the rising edge and the other one on the falling edge. In PWP event action offers, the *T* is captured into CC1 and *tp* into CC0.

The Timer/Counter Event (TCE) n Invert Enable bit in Event Control register (EVCTRL.TCEINVn,where n=0,1 and input events are TCCx_EV_0 or TCCx_EV_1) is used for event source n to select whether the reload should occur on the rising edge or the falling edge. If EVCTRL.TCEINVn=1, the reload will happen on the falling edge.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

The corresponding capture is done only if the channel is enabled in Capture mode (CTRLA.CPTENy=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.MCy is still set, the new timestamp will not be stored and INTFLAG.ERR (INTFLAG<3>) will be set.

**Note:** When using PPW/PWP mode to capture the period and duty cycle of the frequency from the external pin with TCC, one must use EIC and Event System.

**Note:** When up-counting (CTRLBCLR.DIR(CTRLBCLR <0>)=1), counter values lower than 1 cannot be captured in Capture Minimum mode (FCTRLA.CAPTURE(FCTRLA<5:4>)=CAPTMIN or FCTRLB.CAPTURE(FCTRLA<5:4>)=CAPTMIN). To capture the full range including value 0, the TCC must be configured in Down-counting mode (CTRLBSET.DIR(CTRLBCLR <0>)=1).

## 38.5.3. Additional Features

### 38.5.3.1. One-Shot Operation

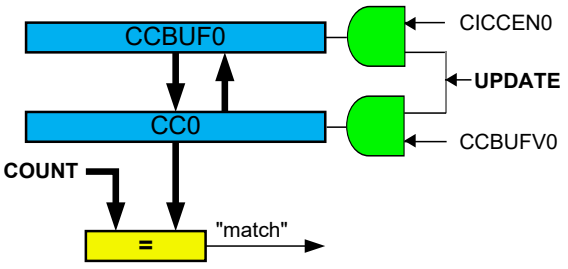
When one-shot is enabled, the counter automatically stops on the next Counter Overflow or Underflow condition. When the counter is stopped, the Stop bit in the Status register (STATUS.STOP(STATUS <0>)) is set and the waveform outputs are set to the value defined by DRVCTRL.NREy and DRVCTRL.NRVy.

One-shot operation can be enabled by writing a '1' to the One-Shot bit in the Control B Set register (CTRLBSET.ONESHOT(CTRLBSET<2>)) and disabled by writing a '1' to CTRLBCLR.ONESHOT(CTRLBCLR<2>). 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(STATUS <0>) is automatically cleared.

### 38.5.3.2. Circular Buffer

The Period register (PER) and the registers (CC0 to CC5) for Compare Channels support circular buffer operation. When circular buffer operation is enabled, the PER or CCy values are copied into the corresponding buffer registers at each update condition. Circular buffering is dedicated to RAMP2, RAMP2A, and DSBOTH operations.

**Figure 38-17.** Circular Buffer on Channel 0




### 38.5.3.3. Ramp Operations

This device supports Five Ramp Operations. All these RAMP Operations require the timer/counter running in single-slope PWM generation operation. The Ramp Operation is selected by writing to the RAMP[2:0] bits (WAVE <6:5>) in the Waveform Control register. The Ramp Operations are broadly divided into RAMP1 Operation and RAMP2x Operation.

#### RAMP1 Operation

This is the default PWM operation, described in *Single-Slope PWM Generation*.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### RAMP2x Operation

These operations are dedicated for power factor correction (PFC), Half-Bridge and Push-Pull SMPS topologies, where two consecutive timer/counter cycles are interleaved, see *Standard RAMP2 Operation*. These cycles are called Ramp A and Ramp B. In the Ramp A cycle, the odd channel output is disabled, and in the Ramp B cycle, the 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(CTRLBSET <4:3>)).

The RAMP2x functionality interleaves the duty cycle of two output signals in a single TCC module. The RAMP2x functions require that the TCC module is used in single slope operation (counting up or down but not both). Each of the different RAMP2x operations use different resources available in the TCC module to achieve the interleaving output.

The TCC module is extremely flexible and can be used in many different types of applications.

Most commonly, the TCC is known for its usage in all forms of motor control applications. In motor control applications, where power is supplied by an AC source there is the need to condition power before the motor control in the system. The TCC module has built in features to enable Interleaved Power Factor Correction to reduce inefficiencies and cost in the system design for power conditioning where an electric motor is controlled. Interleaved Power Factor Correction uses two outputs of a TCC module in RAMP2x configuration to operate two parallel PFC converters at 180º out of phase from each other. It is important to maintain the phase of the two converters precisely to maintain equal load sharing between the two channels. This can all be done simply with the use of the TCC module in one of the three RAMP2x operations. Faults can be generated anywhere in the microcontroller and quickly relayed to the TCC module for immediate handling for protection or control methods.

There are four RAMP2x operations:

1. Standard RAMP2 Operation
2. Alternate RAMP2 (RAMP2A) Operation
3. Critical RAMP2 (RAMP2C) Operation
4. Critical Swapped RAMP2 (RAMP2CS) Operation

### Standard RAMP2 (RAMP2) Operation

RAMP2 operation uses the PER register of the TCC module to determine the period of both output waveforms. CC0 and CC1 registers of the TCC module are used to define the duty cycle of W[0] and W[1] respectfully. To identify the cycle (A or B) of the operation, the Ramp index bit of the STATUS register STATUS.IDX (STATUS<1>) can be read.

In the following figure, the output of both WO[0] and WO[1] is going high (duty cycle begins) when the respective CC0 and CC1 value match COUNT, and goes low (duty cycle ends) when the TOP, or PER value matches the COUNT register value. This is a result of the polarity bits WAVE.POL0 (WAVE<16>) and WAVE.POL1 (WAVE<17>) being set to one. If the polarity bits were cleared, the duty cycle would begin at the reset of COUNT and conclude on the CC0 and CC1 match.

RAMP2 operation can also provide different periods for cycle A (Ramp A) and cycle B (Ramp B). To enable two different period values (TOP) for cycle A and cycle B, the period circular buffer will have to be enabled (WAVE.CIRPEREN (WAVE<7>)=1) as well as writing the PERBUF register. In this operation the following registers are used.

- PER – Period for both cycle A and cycle B when period circular buffer is not enabled
- PERBUF – At initialization, if the period circular buffer is enabled by setting WAVE.CIRPEREN (WAVE<7>) bit (WAVE<7>), then PERBUF defines cycle B period and PER defines cycle A period
- CC0 – Defines the duty cycle of cycle A (Ramp A)
- CC1 – Defines the duty cycle of cycle B (Ramp B)

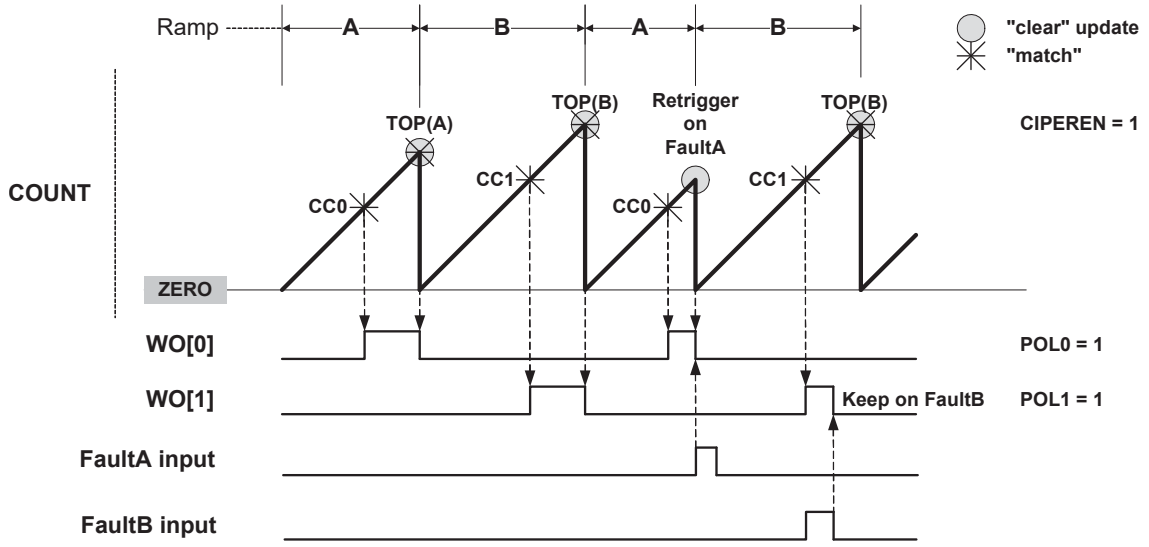

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

In all cases updates to the period should be applied through the period buffer register, PERBUF. The use of the Ramp index bit STATUS.IDX (STATUS<1>) should be used to identify the cycle (A or B) of the operation to be sure the proper period is updated. The following steps are performed to operate the TCC in this operation and initialize the RAMP2 operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the "Peripheral Bus Clock Enable Maskn Register" register of the MCLK module
2. Enable Generic clock for TCC Instance.
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2 by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x1.
6. If cycle B requires different period value than cycle A, enable the Circular buffer by setting WAVE.CIPEREN bit (WAVE<7>).
7. Load the selected Compare/Capture (CC0) register with the desired PWM duty cycle value for Cycle A.
8. Load the another Compare/Capture (CC1) register with the desired PWM duty cycle value for Cycle B.
9. Load the period register PER<31:0> with the desired time period value (TOP) for cycle A.
10. If the circular buffer is enabled, load the period register PERBUF<31:0> with the desired time period value (TOP) for cycle B.
11. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
12. In this operation the output polarity of a selected output signal (if needed) should be set as follow
    a. Set WAVE.POL0 bit (WAVE<16>) to invert WO[0] output.
    b. Set WAVE.POL1 bit (WAVE<17>) to invert WO[1] output
13. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if compare match interrupt is used, set INTENSET.MC0 bit (INTENSET<16> and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
14. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
15. For information on how to clear interrupts, see Interrupts.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-18.** RAMP2 Standard Operation




## Alternate RAMP2 (RAMP2A) Operation

In RAMP2A operation, CC0 is required to use the circular buffer to provide two different duty cycles for cycle A (Ramp A) and cycle B (Ramp B). Updates to the duty cycle can be made by writing the CCBUF0 register at the time of an INTFLAG.OVF (INTFLAG<0>) or INTFLAG.MCy flags flag or MC flag. Writing CCBUF0 at the time of an INTFLAG.OVF (INTFLAG<0>) flag will change the duty-cycle of cycle B. Writing CCBUF0 at the time of an INTFLAG.MCy flag will change the duty cycle of cycle A. To further identify the cycle (A or B) of the operation refer to the Ramp index bit of the STATUS register, STATUS.IDX bit (STATUS<1>). To eliminate the need of monitoring the INTFLAG.OVF (INTFLAG<0>) or INTFLAG.MCy flags for the update to the duty-cycle, the DMA can be used to populate the appropriate buffer register.

RAMP2A operation uses the PER register of the TCC module to determine the period of both output waveforms. If different periods are needed for cycle A and cycle B outputs, the period circular buffer (CIPEREN) will have to be enabled. As mentioned for the duty cycle, the PER and PERBUF registers can be treated in the same manner as the CC0 and CCBUF0 to provide the period updates at the appropriate times. If the DMA is to be used, a separate DMA channel will need to be initialized for the period management.

- PER – Defines the period for both cycle A and cycle B if the period circular buffer is NOT enabled
- PERBUF – At startup PERBUF defines cycle B period and PER defines cycle A period. After the TCC module is started PERBUF holds the period value (TOP) of the next cycle
- CC0 – Defines the duty cycle for both cycle A and cycle B if the CC circular buffer is NOT enabled
- CCBUF0 – At startup CCBUF0 defines cycle B duty cycle and CC0 defines cycle A duty cycle. After the TCC module is started CCBUF holds the duty cycle value of the next cycle

In all cases for use with or without DMA in RAMP2A operation, updates to the period or duty cycle should be applied through the buffer registers (PERBUF or CCBUFy). If both the period and duty cycle have different values in cycle A and cycle B and the DMA is used, two DMA channels will be needed, one for PERBUF and one for CCBUFy. Refer to *RAMP2 Alternate Operation*.

The following steps are performed to operate the TCC in this mode.

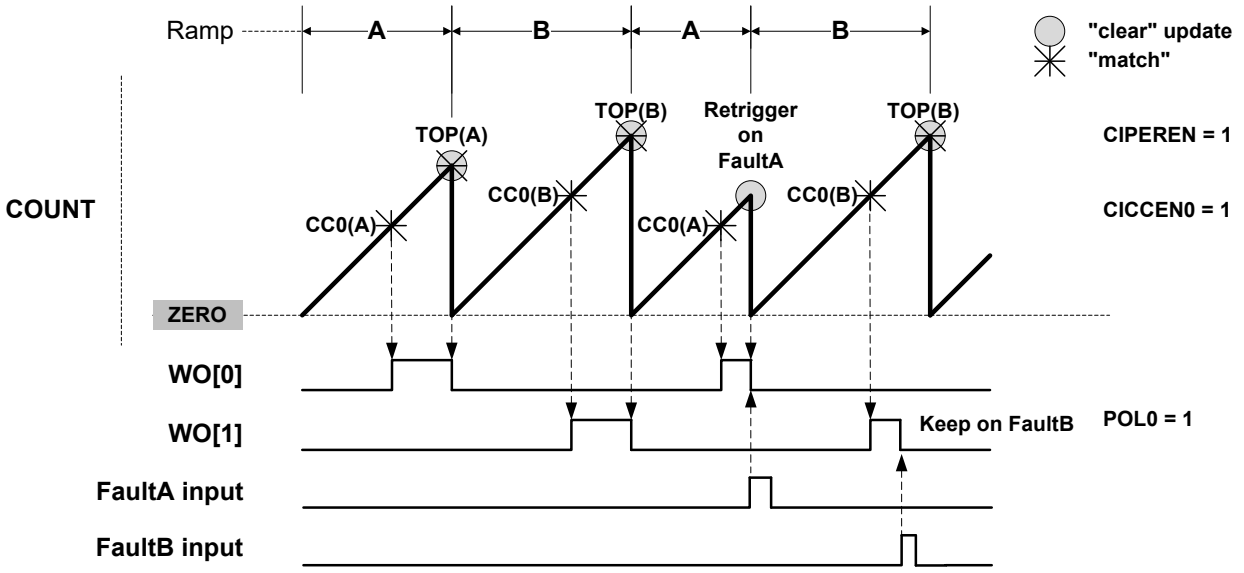
1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the "Peripheral Bus Clock Enable Maskn Register" register of the MCLK module
2. Enable Generic clock for TCC Instance.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2A by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x2.
6. If cycle A requires different period value than cycle B, enable the Circular buffer for period by setting WAVE.CIPEREN bit (WAVE<7>).
7. Load Compare/Capture (CC0<31:0>) register with the desired PWM duty cycle value for Cycle A.
8. Load Compare/Capture buffer (CCBUF0<31:0>) register with the desired PWM duty cycle value for Cycle B.
9. Load the period register PER<31:0> with the desired time period value (TOP) for cycle A.
10. Load the period register PERBUF<31:0> with the desired time period value(TOP) for cycle B.
11. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
12. In this operation the output polarity of both output signals on WO[0] and WO[1], is controlled by setting WAVE.POL0 bit (WAVE<16>)
13. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if the compare match interrupt is used, set INTENSET.MC0 bit (INTENSET<16>) and configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
14. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
15. For information on how to clear interrupts, see Interrupts.

**Note:** Retrigger in RAMP2 operations (RAMP2, RAMP2A, RAMP2C, RAMP2CS) is not supported if a prescaler is used (CTRLA.PRESCALER != 0) and the retrigger of the counter is done on the next GCLK (CTRLA.PRESCSYNC = GCLK or CTRLA.PRESCSYNC = RESYNC). If use of the prescaler is required, configure the retrigger of the counter on the next prescaler clock (CTRLA.PRESCSYNC = PRESC).

**Figure 38-19.** 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 operation, 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

on WO[0] will last the entire period of Ramp A, if WAVE.POL0 bit (WAVE<16>)=0. Refer to *RAMP2 Critical Operation With More Than 2 Channels* for more information.

The RAMP2C operation is the simplest to setup and understand as circular buffers are not used in the implementation. The period and duty cycle of output is controlled as shown below.

- PER – Defines the period for cycle B (Ramp B)
- CC0 – Defines the period for cycle A (Ramp A)
- CC1 – Defines the duty cycle for cycle B (Ramp B)
- CC2 – Defines the duty cycle for cycle A (Ramp A)

*RAMP2 Critical Operation With More Than 2 Channels* illustrates the RAMP2CS operation with three active channels. The following steps are performed to operate the TCC in this operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the "Peripheral Bus Clock Enable Maskn Register" register of the MCLK module.
2. Enable Generic clock for TCC Instance.
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2C by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x3.
6. Load Compare/Capture (CC1<31:0>) register with the desired PWM duty cycle value for Cycle B.
7. Load Compare/Capture buffer register (CC2<31:0>) register with the desired PWM duty cycle value for Cycle A.
8. Load the period register PER<31:0> with the desired time period value (TOP) for cycle B.
9. Load the Compare/Capture register (CC0<31:0>) with the desired time period value (TOP) for cycle A.
10. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
11. In this operation the output polarity of a selected output signal should be set as follow
    a. Set WAVE.POL2 bit (WAVE<18>) as this controls WO[0] output.
    b. Set WAVE.POL1 bit (WAVE<17>) as this controls WO[1] output
12. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>) ,if compare match interrupt is used, set INTENSET.MC2 bit (INTENSET<18>) for Ramp A match, INTENSET.MC1 bit (INTENSET<17>) for Ramp B match interrupt. Configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
13. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
14. For information on how to clear interrupts, see Interrupts.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-20.** RAMP2 Critical Operation With More Than 2 Channels

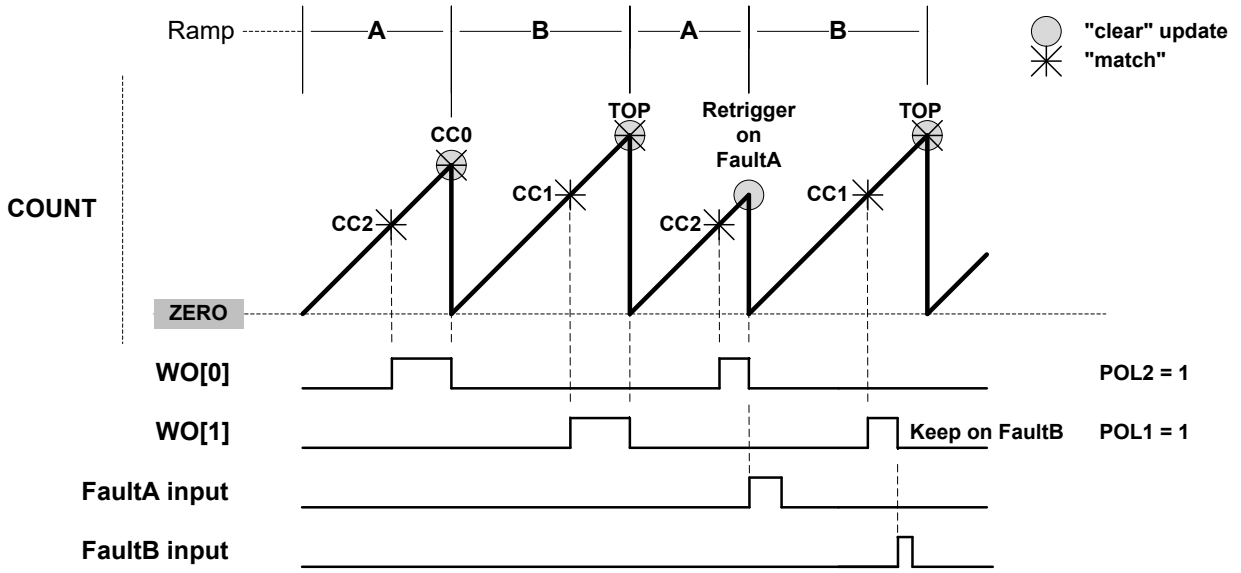
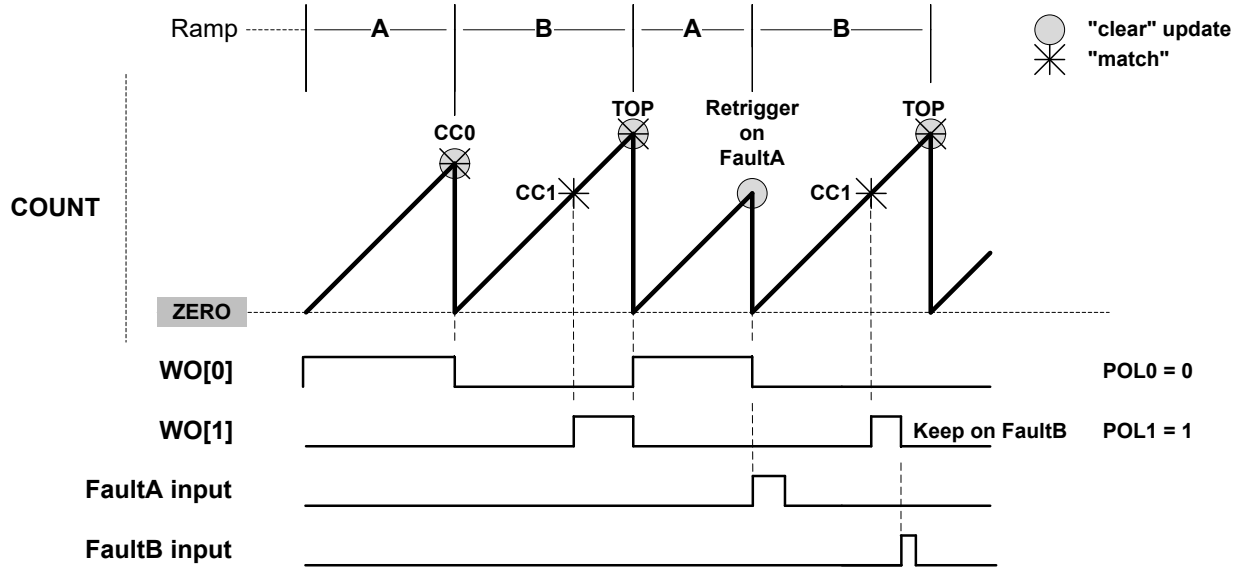



**Figure 38-21.** RAMP2 Critical Operation With 2 Channels




### Critical Swapped RAMP2 (RAMP2C) Operation

In RAMP2CS variant, WO[0] and W0[1] active ramp are inverted, WO[0] is active on RAMPB and WO[1] is active on RAMPA. The RAMP2CS operation is similar to RAMP2C operation and does not use circular buffers in the implementation. The period and duty cycle of output is controlled as shown below.

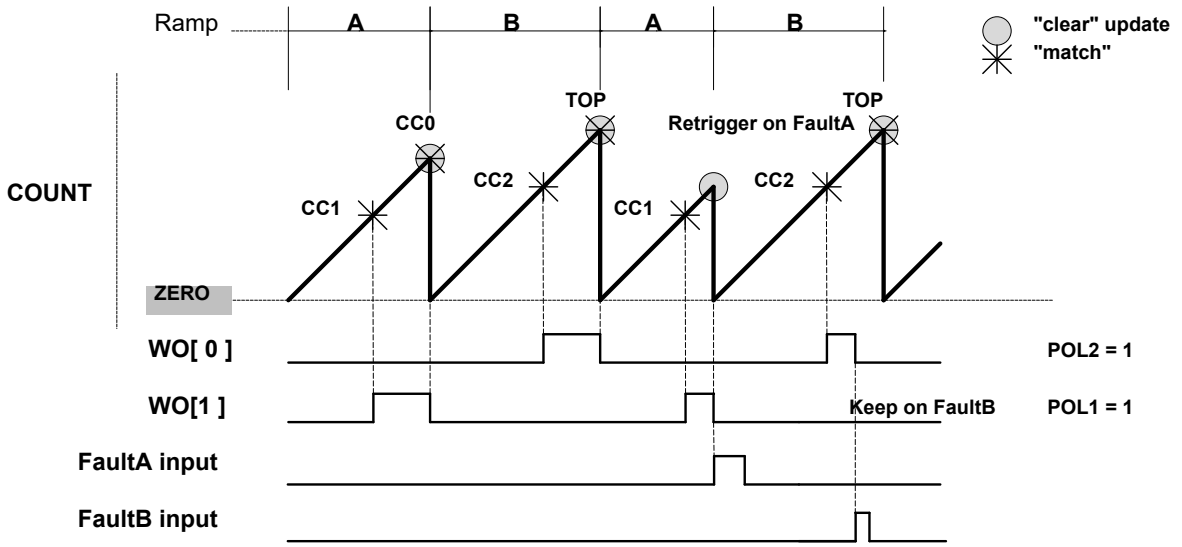
- PER – Defines the period for cycle B (Ramp B)
- CC0 – Defines the period for cycle A (Ramp A)
- CC1 – Defines the duty cycle for cycle A (Ramp A)
- CC2 – Defines the duty cycle for cycle B (Ramp B)


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

RAMP2 Critical Swap Operation illustrates the RAMP2CS operation. The following steps are performed to operate the TCC in this operation.

1. Configure the clock source for the TCC Instance in the Main Clock Controller (MCLK) and enable the APB BUS clock for the TCC Instance by writing a '1' to the TCCx_ bit in the "Peripheral Bus Clock Enable Maskn Register" register of the MCLK module
2. Enable Generic clock for TCC Instance.
3. Select desired prescaler by setting CTRLA.PRESCALER bits (CTRLA<10:8>).
4. Set waveform generation operation to Normal Pulse Width Modulation Operation (NPWM) by setting WAVE.WAVEGEN bits (WAVE<2:0> = 2).
5. Set Ramp operation to RAMP2CS by setting value of WAVE.RAMP bits (WAVE<6:4>) to 0x4.
6. Load Compare/Capture (CC1<31:0>) register with the desired PWM duty cycle value for Cycle A (Ramp A).
7. Load Compare/Capture buffer register (CC2<31:0>) register with the desired PWM duty cycle value for Cycle B (Ramp B).
8. Load the period register PER<31:0> with the desired time period value (TOP) for cycle B.
9. Load the Compare/Capture register (CC0<31:0>) with the desired time period value (TOP) for cycle A.
10. Set Counter to count in up direction by clearing CTRLBCLR.DIR bit (CTRLBCLR<0> = 1).
11. In this operation the output polarity of a selected output signal should be set as follows:
    a. Set WAVE.POL2 bit (WAVE<18>) as this controls WO[0] output.
    b. Set WAVE.POL1 bit (WAVE<17>) as this controls WO[1] output
12. If overflow interrupt is used, set INTENSET.OVF bit (INTENSET<0>), if compare match interrupt is used, set INTENSET.MC1 bit (INTENSET<17>) for Ramp A match, INTENSET.MC2 bit (INTENSET<18>) for Ramp B match interrupt. Configure the NVIC by setting group priority, sub priority and enabling corresponding TCCx IRQ.
13. Enable TCC by setting CTRLA.ENABLE bit (CTRLA<1>).
14. For information on how to clear interrupts, see Interrupts.

**Figure 38-22.** RAMP2 Critical Swap Operation





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.5.3.4. 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 (TCCx_MC_0 and TCCx_MC_1) 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 operation.

#### Fault Filtering

There are three filters available for both Fault A and Fault B input. They are configured by the corresponding Recoverable Fault Configuration registers (FCTRLA and FCTRLB). The three filters can either be used independently or in any combination.

**Input Filtering**

When disabled (FCTRLx.FILTERVAL = 0), the action of a fault on the output port is asynchronous. This helps the fault action to be performed on the output of the compare channel, even when the fault is caused by the loss of the system clock. Therefore, when a fault occurs, the system will immediately and asynchronously disable the compare channel output as long as a fault is present. To avoid TCC control block corruption by a glitch on a fault input line, the fault action on the TCC control block (retrigger, capture, etc..) is synchronized on internal clock domain. A digital filter can be enabled and configured by the Fault Filter Value bits in the Fault Configuration registers (FCTRLA.FILTERVAL (FCTRLA <28:24>) and FCTRLB.FILTERVAL (FCTRLB <28:24>)). If the fault width is less than the FILTERVAL (in TCC clock cycles), will be ignored. A valid fault event action on the TCC FSM, will be then delayed by the clock cycles represented by the FILTERVAL.

**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 A or Fault B Blanking Mode bits in the Recoverable Fault Configuration registers (FCTRLA.BLANK (FCTRLA <6:5>) or FCTRLB.BLANK (FCTRLB <6:5>)). The desired duration of the blanking must be written to the Fault Blanking Time bits (FCTRLA.BLANKVAL (FCTRLA <23:16>) or FCTRLB.BLANKVAL (FCTRLB <23:16>)). The blanking time $t_b$ is calculated by

$$t_b = \frac{1 + \mathrm{BLANKVAL}}{f_{\mathrm{GCLK_TCCx_PRESC}}}$$

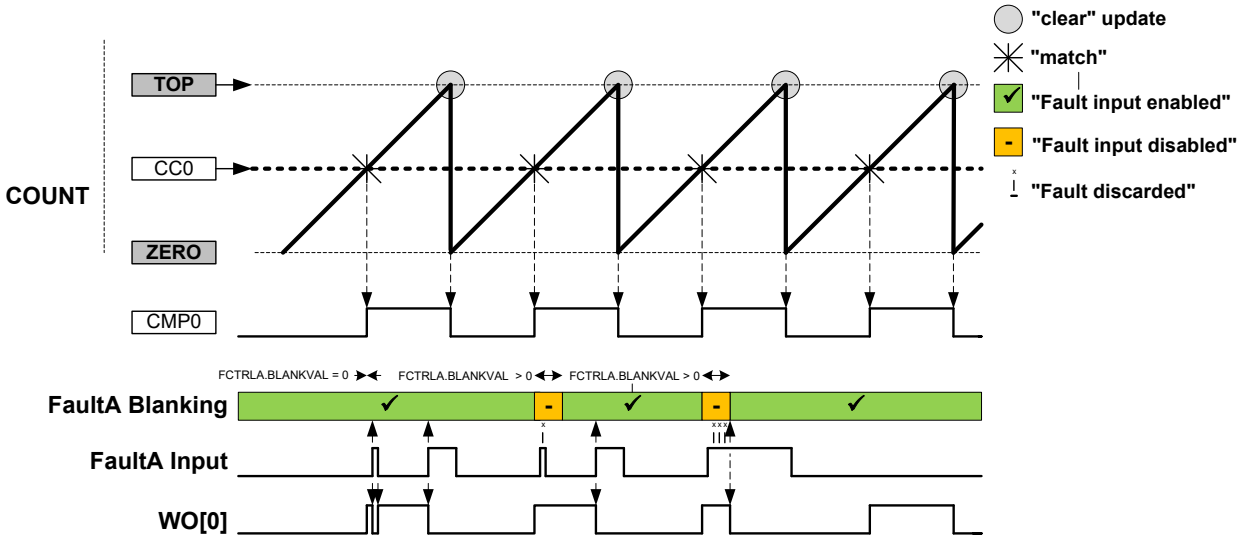
Here, $f_{\mathrm{GCLK_TCCx_PRESC}}$ is the frequency of the prescaled peripheral clock frequency $f_{\mathrm{GCLK_TCCx}}$.

The prescaler is enabled by writing '1' to the Fault A or Fault B Blanking Prescaler bit (FCTRLA.BLANKPRESC(FCTRLA <15>) or FCTRLB.BLANKPRESC(FCTRLB <15>)). 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 (FCTRLA.BLANKVAL (FCTRLA <23:16>) or FCTRLB.BLANKVAL (FCTRLB <23:16>))=
255) at $f_{\mathrm{GCLK_TCCx}}$=96MHz is 2.67µs (no prescaler) or 170µs (prescaling). For $f_{\mathrm{GCLK_TCCx}}$=1MHz, the maximum blanking time is either 170µs (no prescaling) or 10.9ms (prescaling enabled).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

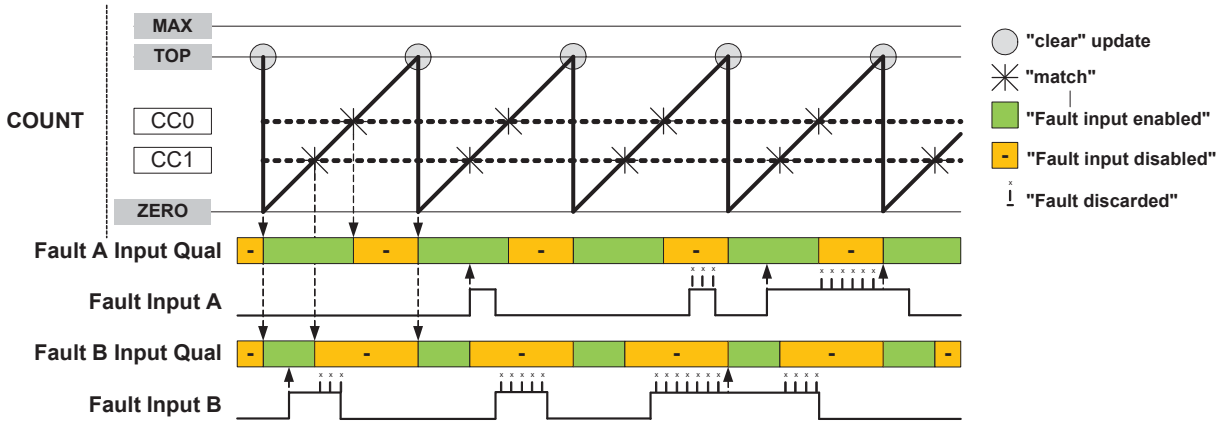
**Figure 38-23.** Fault Blanking in RAMP1 Operation with Inverted Polarity




**Fault Qualification**

This is enabled by writing a '1' to the Fault A or Fault B Qualification bit in the Recoverable Fault A or Fault B Configuration register ((FCTRLA.QUAL(FCTRLA <4>) or FCTRLB.QUAL(FCTRLB <4>)). When the recoverable fault qualification is enabled (FCTRLA.QUAL(FCTRLA <4>) =1 or FCTRLB.QUAL(FCTRLB <4>)=1, the fault input is disabled all the time and the corresponding channel output has an inactive level, as shown in the figures below.

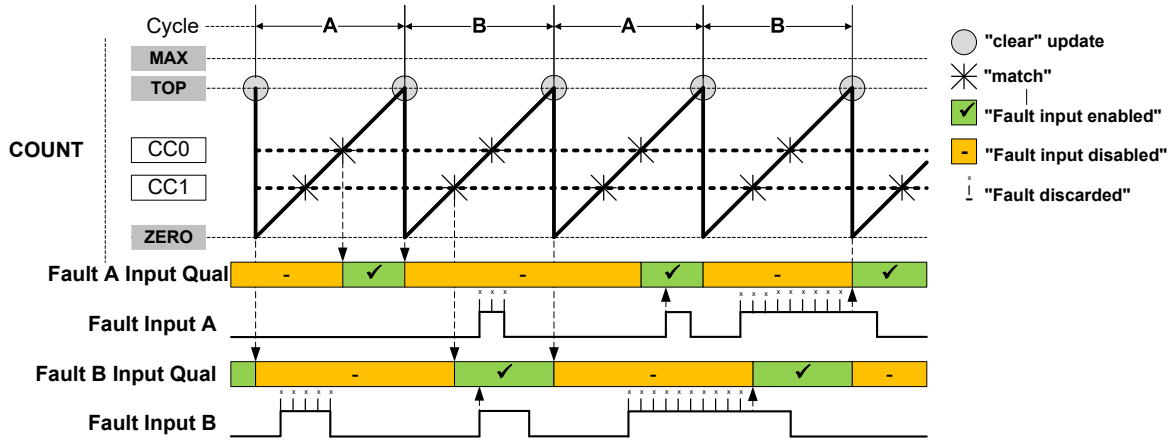
**Figure 38-24.** Fault Qualification in RAMP1 Operation





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-25.** Fault Qualification in RAMP2 Operation with Inverted Polarity

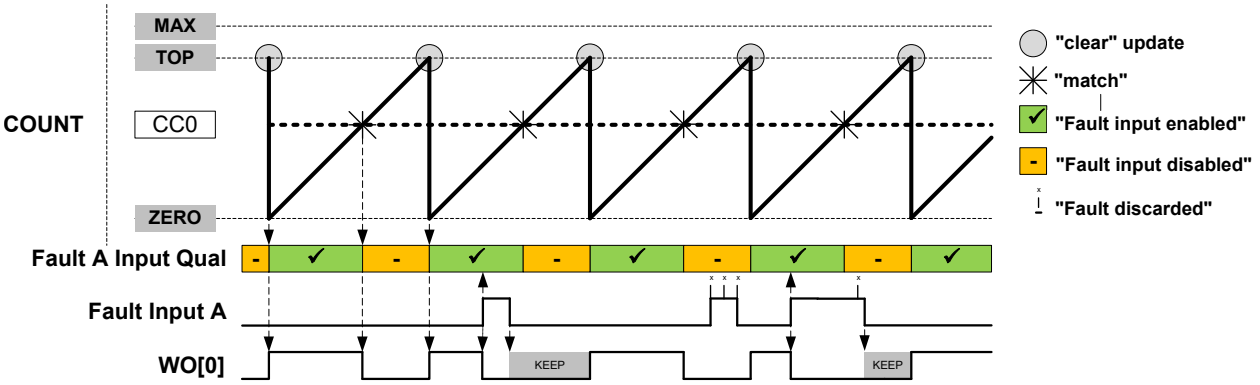



## Fault Actions

Different fault actions can be configured individually for Fault A and Fault B. Most fault actions are not mutually exclusive; therefore 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 A or Fault B Keeper bit in the corresponding Recoverable Fault A or Fault B Configuration register (FCTRLA.KEEP(FCTRLA <3>) or FCTRLB.KEEP(FCTRLB <3>) 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.

**Figure 38-26.** Waveform Generation with Fault Qualification and Keep Action




**Restart Action** This is enabled by writing the Fault A or Fault B Restart bit in corresponding Recoverable Fault A or Fault B Configuration register (FCTRLA.RESTART(FCTRLA <7>) =1 or FCTRLB.RESTART(FCTRLB <7>) 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 *Waveform Generation in RAMP1 Operation with Restart Action*. In RAMP1 operation, 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 *Waveform Generation in RAMP2 Operation with Restart Action*. 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-27.** Waveform Generation in RAMP1 Operation with Restart Action

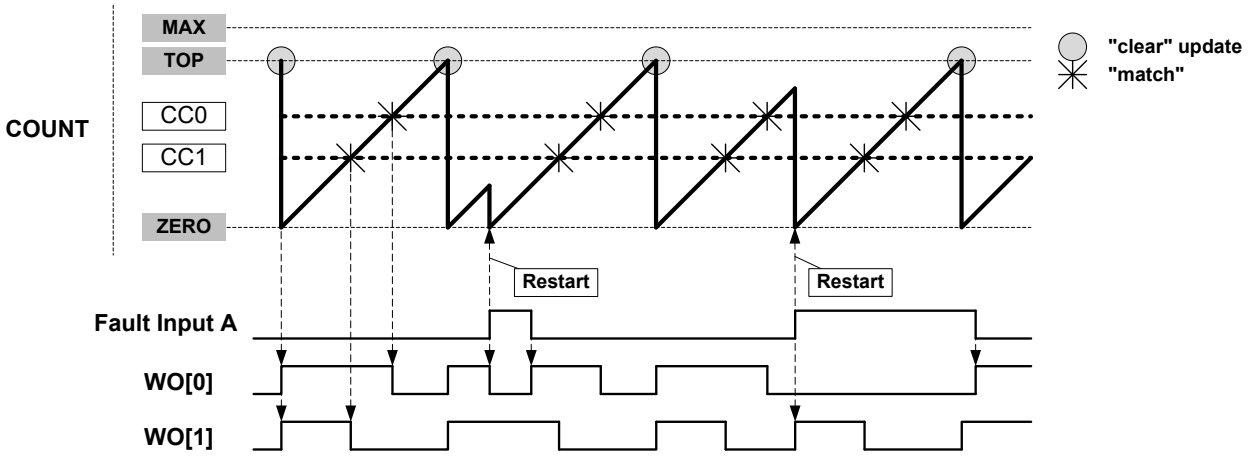
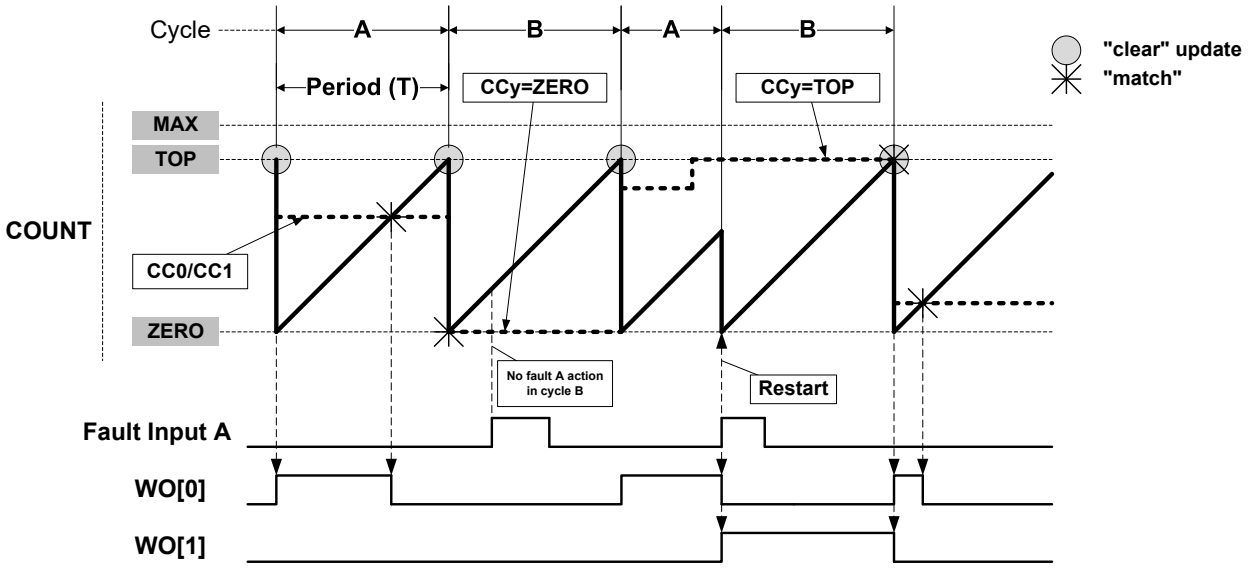



**Figure 38-28.** Waveform Generation in RAMP2 Operation with Restart Action




**Capture Action**

Several capture actions can be selected by writing the Fault A or Fault B Capture Action bits in the Fault A or Fault B Control register (FCTRLA.CAPTURE(FCTRLA <14:12>) or FCTRLB.CAPTURE(FCTRLB <14:12>). When one of the capture operations is selected, the counter value is captured when the fault occurs. The following capture operations are available:

- CAPT - the equivalent to a standard capture operation, for further details 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 *Capture Action "CAPTMAX"* in the FCTRLA or FCTRLB register.
- 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- DERIV0 - notifies by event or interrupt when a local extreme captured value is detected, see *Capture Action "DERIV0"* in the FCTRLA or FCTRLB register.

*CCy Content:*

In CAPTMIN and CAPTMAX operations, CCy keeps the respective extreme captured values, see *Capture Action "CAPTMAX"*. In LOCMIN, LOCMAX or DERIV0 operation, CCy follows the counter value at fault time, see *Capture Action "DERIV0"*.

Before enabling CAPTMIN or CAPTMAX mode of capture, the user must initialize the corresponding CCy register value to a value different from zero (for CAPTMIN) and TOP (for CAPTMAX). If the initial value of the CCy register is zero for CAPTMIN and TOP for CAPTMAX, no captures will be performed using the corresponding channel.

*MCy Behaviour:*

In LOCMIN and LOCMAX operation, capture is performed on each capture event. The MCy interrupt flag is set only when the captured value is above or equal for LOCMIN and below or equal for LOCMAX to the previous captured value. So 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).

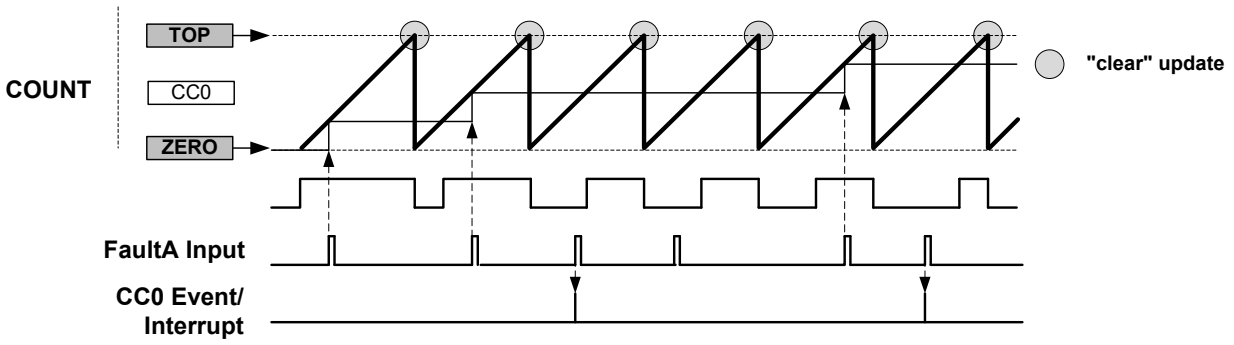
In CAPT operation, capture is performed on each capture event. The MCy interrupt flag is set on each new capture.

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 MCy interrupt flag is set only when on capture event time, the counter value is greater than or equal (for CAPTMIN), or less than 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 A and Fault B and each dedicated CCy channel capture counter value. In other modes, an interrupt is only generated on an extreme captured value.

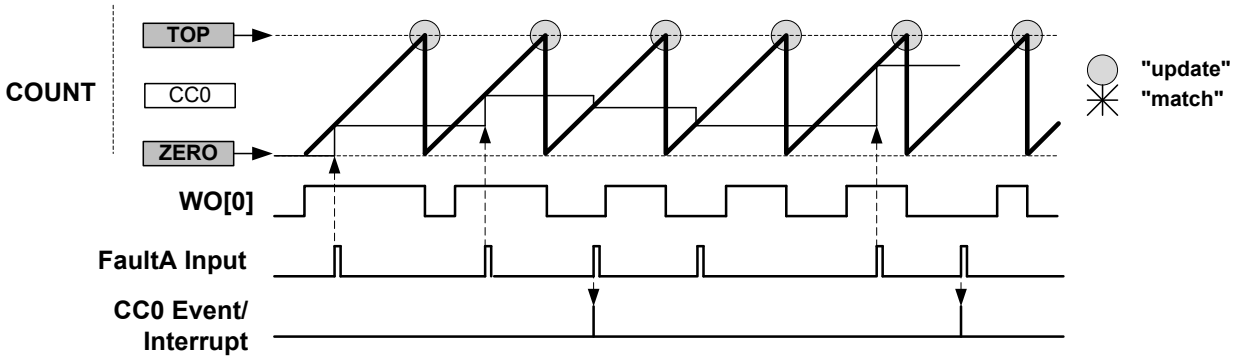
**Figure 38-29.** Capture Action "CAPTMAX"





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-30.** Capture Action “DERIV0”




**Hardware Halt Action**

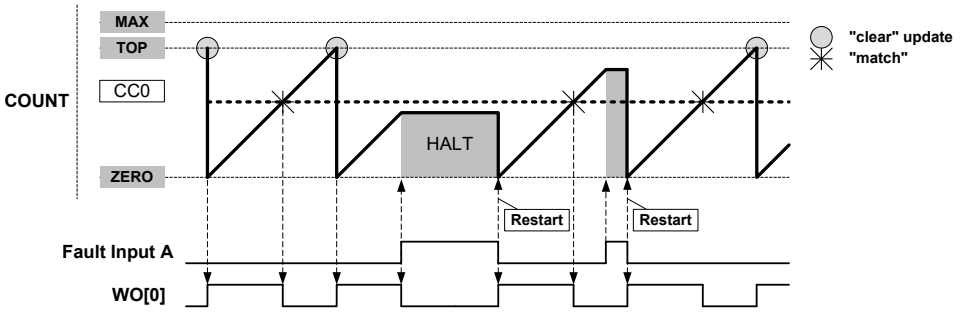
This is configured by writing 0x1 to the Fault A or Fault B Halt mode bits in the Recoverable Fault A or Fault B Configuration registers (FCTRLA.HALT(FCTRLA <9:8>) or FCTRLB.HALT(FCTRLB <9:8>). When enabled, the timer/counter is halted and the cycle is extended as long as the corresponding fault is present.

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.

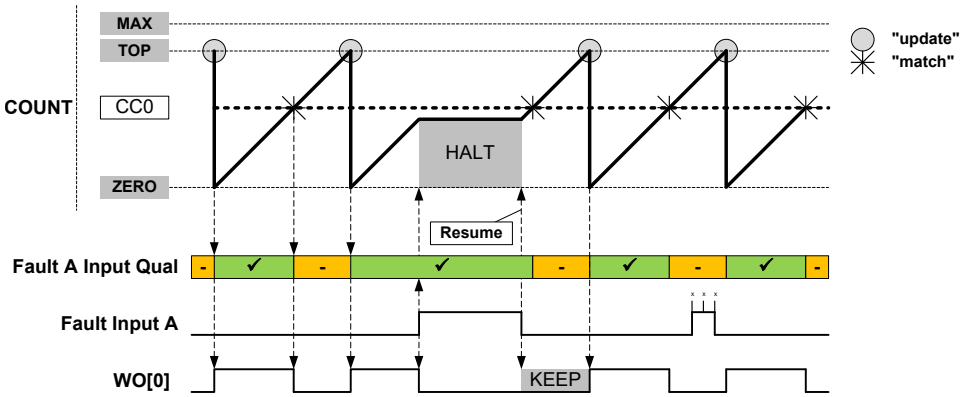
**Figure 38-31.** Waveform Generation with Halt and Restart Actions





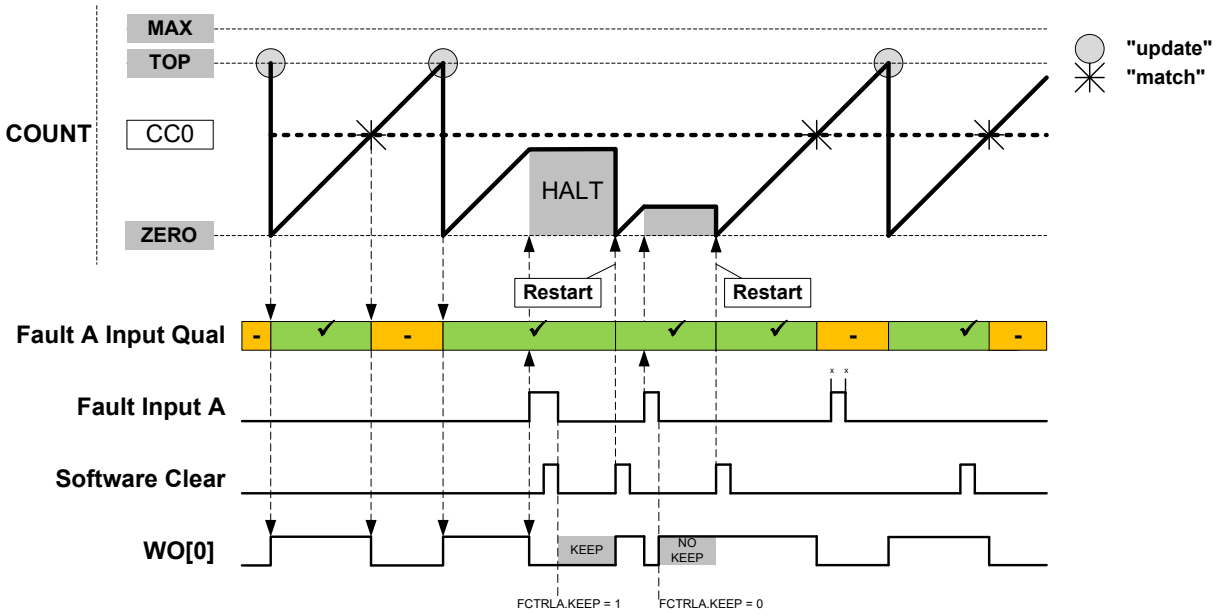
© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 38-32.** Waveform Generation with Fault Qualification, Halt, and Restart Actions




**Software Halt Action** This action is configured by writing 0x2 to the Fault A and Fault B Halt mode bits in the Recoverable Fault A or Fault B configuration registers (FCTRLA.HALT(FCTRLA <9:8>) or FCTRLB.HALT(FCTRLB <9:8>). 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 A or Fault B bit in the STATUS register must be cleared by software. See the following figure.

**Figure 38-33.** Waveform Generation with Software Halt, Fault Qualification, Keep and Restart Actions




### 38.5.3.5. 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.NREy and DRVCTRL.NRVy). The non-recoverable fault input (TCCx_EV_0 and TCCx_EV_1) actions are enabled in Event Control register (EVCTRL.EVACT0(EVCTRL <2:0>) and EVCTRL.EVACT1(EVCTRL <5:3>)).

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 Filter Value bits in the Driver Control register DRVCTRL.FILTERVAL0 (DRVCTRL<27:24>) and DRVCTRL.FILTERVAL1 (DRVCTRL<31:28>) . Therefore, the event detection is synchronous, and event action is delayed by the selected digital filter value clock cycles.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

When the Fault Detection on Debug Break Detection bit in Debug Control register (DGBCTRL.FDDBD (DGBCTRL <2>)) 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 (CTRLBSET <1>)=1 and the Ramp index or counter direction changes, a non-recoverable Update Fault State and the respective interrupt (UFS) are generated.

#### 38.5.3.6. Time-Stamp Capture

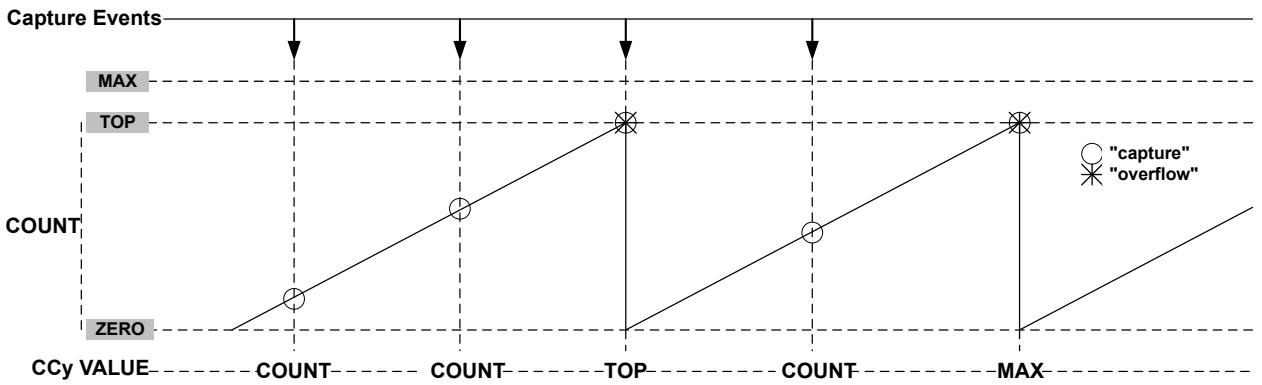
This feature is enabled when the Capture Time Stamp (STAMP) Event Action in Event Control register (EVCTRL.EVACT0(EVCTRL <2:0>)=0x6) is selected. The counter TOP value must be smaller than MAX.

When a capture event is detected, the COUNT value is copied into the corresponding Channel y Compare/Capture Value (CCy) register. In case of an overflow, the MAX value is copied into the corresponding CCy register.

When a valid captured value is present in the capture channel register, the corresponding Capture Channel y Interrupt Flag (INTFLAG.MCy) 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.MCy) is still set, the new time-stamp will not be stored and INTFLAG.ERR will be set.

**Figure 38-34.** Time-Stamp




### 38.5.4. DMA, Interrupts, and Events

The following table shows the DMA request for the TCC module.

**Table 38-4.** 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 CCy register is read |
| Retrigger | Yes | Yes | | | |
| Count | Yes | Yes | | | |
| Capture Overflow Error | Yes | | | | |
| Debug Fault State | Yes | | | | |
| Recoverable Faults | Yes | | | | |
| Non-Recoverable Faults | Yes | | | | |
| TCCx Event 0 input | | | Yes(4) | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-4.** Module Requests for TCC (continued)

| Condition | Interrupt request | Event output | Event input | DMA request | DMA request is cleared |
|---|---|---|---|---|---|
| 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

#### 38.5.4.1. 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(CTRLA<23>)) 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(CTRLA<23>)=1, the TCC generates a DMA trigger on the cycle following the DMA One-Shot Command written to the Control B register (CTRLBSET.CMD(CTRLBSET<7:5>)=DMAOS).<br>In both cases, the request is cleared by hardware on DMA acknowledge. |
| **Channel Match (MCy)** | A DMA request is set only on a compare match if CTRLA.DMAOS(CTRLA<23>)=0. The request is cleared by hardware on DMA acknowledge.<br>When CTRLA.DMAOS(CTRLA<23>)=1, the DMA requests are not generated. |
| **Channel Capture (MCy)** | For a capture channel, the request is set when valid data is present in the CCy register, and cleared once the CCy register is read.<br>In this operation mode, the CTRLA.DMAOS bit (CTRLA<23>) value is ignored. |

**Important:** DMA One-Shot mode is not available in RAMP1/RAMP2C/RAMP2CS modes.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 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.

**Note:** Circular buffer are intended to be used with RAMP2, RAMP2A and DSBOTH operation only.

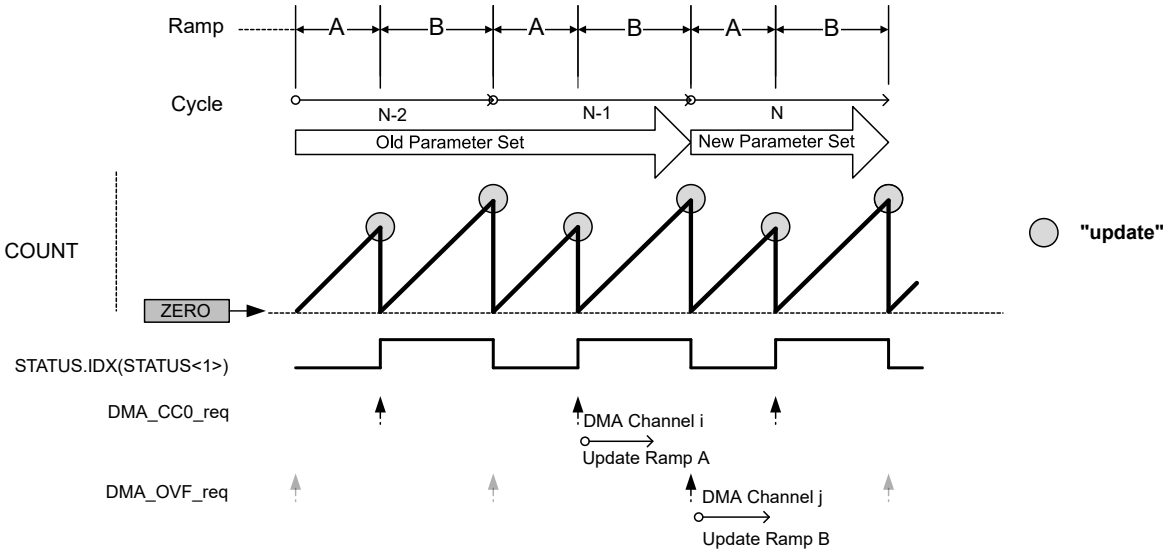
*DMA Operation with Circular Buffer in RAMP2 and RAMP2A Operation*

When a CC0 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 (Compare Match) 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 38-35.** DMA Triggers in RAMP2 Operation and Circular Buffer Enabled




*DMA Operation with Circular Buffer in DSBOTH*

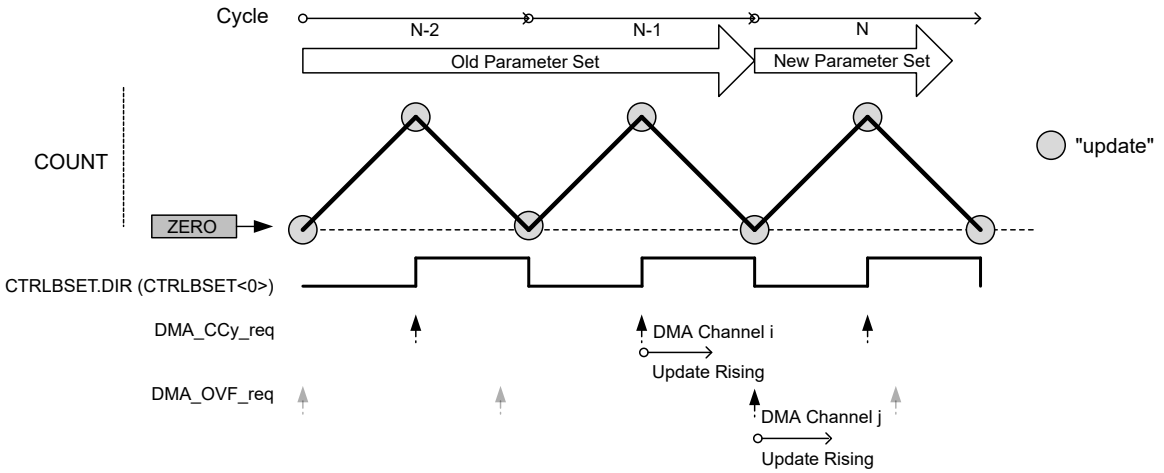
When CC0 channel is selected as a circular buffer, the related DMA request is not set on a compare match detection, but on start of down-counting phase.

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 (Compare Match) 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 38-36.** DMA Triggers in DSBOTH Operation and Circular Buffer Enabled




### 38.5.4.2. Interrupts

The TCC has the following interrupt sources (refer to the INTENSET register):

- 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 (FAULTA and FAULTB)
- Non-recoverable Faults (FAULT0 and FAULT1)
- Compare Match or Capture Channels (MCy)

These interrupts are asynchronous wake-up sources. See Sleep Mode Entry and Exit Table in PM/ Sleep Mode Controller section for details.

References:

- PM_Sleep Mode Controller

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. Refer to the INTFLAG for details on how to clear Interrupt flags. The TCC has one common interrupt request line for OVF, ERR, UFS, DFS, FAULTA, FAULTB, FAULT0, and FAULT1 interrupt sources. It contains a separate line for both TRG, and CNT interrupt sources and another line for all Compare Match or Capture Channels (MCy) 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.

References:

- Nested Vector Interrupt Controller


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 38.5.4.3. Events

The TCC can generate the following output events:

- Overflow/Underflow (OVF)
- Trigger (TRG)
- Counter (CNT) For further details, refer to EVCTRL.CNTSEL description.
- Compare Match or Capture on compare/capture channels: MCy

Writing a '1' or '0' to an Event Output bit in the Event Control Register (OVFEO, TRGEO, CNTEO and MCEOy bits) enables or disables the corresponding output event. Refer also to *EVSYS – Event System*.

The TCC can take the following actions on a channel input event (MCy):

- Capture event
- Generate a recoverable or non-recoverable fault

The TCC can take the following actions on counter Event 1 (TCCx_EV_1):

- 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_EV_0):

- 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(EVCTRL <2:0>) and EVCTRL.EVACT1(EVCTRL <5:3>)). For further details, refer to EVCTRL.

Writing a '1' to an Event Input bit in the Event Control register (EVCTRL.MCEIy or EVCTRL.TCEIn where n=0,1) enables the corresponding action on input event, where writing '0' disables the 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.

Reference:

- EVSYS

### 38.5.5. 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 TCC interrupt 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

References:

- EVSYS

## 38.6. Peripheral Dependencies

**Table 38-5.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| TCC0 | 0x44810000 | 88 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS<br>89 : CNT/TRG<br>90 : MC0<br>91 : MC1 | **APB:**<br>CLKMSK[1]<br>MASK8 | PCHCTRL[25] | 30<br>STATUS0<br>PERID30 | 12 : OVF<br>13 : MC0<br>14 : MC1 | Generators:<br>**OVF:**<br>CHANNEL[41]<br>**TRG:**<br>CHANNEL[42]<br>**CNT:**<br>CHANNEL[43]<br>**MC0:**<br>CHANNEL[44]<br>**MC1:**<br>CHANNEL[45]<br>Users:<br>**EV0:**<br>USER[10],Resynch<br>**EV1:**<br>USER[11],Resynch<br>**MC0:**<br>USER[12],Resynch<br>**MC1:**<br>USER[13],Resynch | PD_CORE_SW |
| TCC1 | 0x44812000 | 92 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS<br>93 : CNT/TRG<br>94 : MC0<br>95 : MC1 | **APB:**<br>CLKMSK[1]<br>MASK9 | PCHCTRL[25] | 31<br>STATUS0<br>PERID31 | 15 : OVF<br>16 : MC0<br>17 : MC1 | Generators:<br>**OVF:**<br>CHANNEL[46]<br>**TRG:**<br>CHANNEL[47]<br>**CNT:**<br>CHANNEL[48]<br>**MC0:**<br>CHANNEL[49]<br>**MC1:**<br>CHANNEL[50]<br>Users:<br>**EV0:**<br>USER[14],Resynch<br>**EV1:**<br>USER[15],Resynch<br>**MC0:**<br>USER[16],Resynch<br>**MC1:**<br>USER[17],Resynch | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-5.** Peripheral Dependencies (continued)

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| TCC2 | 0x44814000 | 96 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS<br>97 : CNT/TRG<br>98 : MC0<br>99 : MC1 | **APB:**<br>CLKMSK[1]<br>MASK10 | PCHCTRL[26] | 32<br>STATUS1<br>PERID0 | 18 : OVF<br>19 : MC0<br>20 : MC1 | Generators:<br>**OVF:**<br>CHANNEL[51]<br>**TRG:**<br>CHANNEL[52]<br>**CNT:**<br>CHANNEL[53]<br>**MC0:**<br>CHANNEL[54]<br>**MC1:**<br>CHANNEL[55]<br>Users:<br>**EV0:**<br>USER[18],Resynch<br>**EV1:**<br>USER[19],Resynch<br>**MC0:**<br>USER[20],Resynch<br>**MC1:**<br>USER[21],Resynch | PD_CORE_SW |
| TCC3 | 0x44816000 | 100 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS<br>101 : CNT/TRG<br>102 : MC0<br>103 : MC1 | **APB:**<br>CLKMSK[1]<br>MASK11 | PCHCTRL[27] | 33<br>STATUS1<br>PERID1 | 21 : OVF<br>22 : MC0<br>23 : MC1 | Generators:<br>**OVF:**<br>CHANNEL[56]<br>**TRG:**<br>CHANNEL[57]<br>**CNT:**<br>CHANNEL[58]<br>**MC0:**<br>CHANNEL[59]<br>**MC1:**<br>CHANNEL[60]<br>Users:<br>**EV0:**<br>USER[22],Resynch<br>**EV1:**<br>USER[23],Resynch<br>**MC0:**<br>USER[24],Resynch<br>**MC1:**<br>USER[25],Resynch | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-5.** Peripheral Dependencies (continued)

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| TCC4 | 0x44824000 | 118 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS<br>119 : CNT/TRG<br>120 : MC0<br>121 : MC1 | **APB:**<br>CLKMSK[1]<br>MASK14 | PCHCTRL[30] | 36<br>STATUS1<br>PERID4 | 28 : OVF<br>29 : MC0<br>30 : MC1 | Generators:<br>**OVF:**<br>CHANNEL[61]<br>**TRG:**<br>CHANNEL[62]<br>**CNT:**<br>CHANNEL[63]<br>**MC0:**<br>CHANNEL[64]<br>**MC1:**<br>CHANNEL[65]<br>Users:<br>**EV0:**<br>USER[26],Resynch<br>**EV1:**<br>USER[27],Resynch<br>**MC0:**<br>USER[28],Resynch<br>**MC1:**<br>USER[29],Resynch | PD_CORE_SW |
| TCC5 | 0x44826000 | 122 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS<br>123 : CNT/TRG<br>124 : MC0<br>125 : MC1 | **APB:**<br>CLKMSK[1]<br>MASK15 | PCHCTRL[31] | 37<br>STATUS1<br>PERID5 | 31 : OVF<br>32 : MC0<br>33 : MC1 | Generators:<br>**OVF:**<br>CHANNEL[66]<br>**TRG:**<br>CHANNEL[67]<br>**CNT:**<br>CHANNEL[68]<br>**MC0:**<br>CHANNEL[69]<br>**MC1:**<br>CHANNEL[70]<br>Users:<br>**EV0:**<br>USER[30],Resynch<br>**EV1:**<br>USER[31],Resynch<br>**MC0:**<br>USER[32],Resynch<br>**MC1:**<br>USER[33],Resynch | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-5.** Peripheral Dependencies (continued)

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| TCC6 | 0x44828000 | 126 : DFS/ERR/ FAULTA/ FAULTB/ FAULT0/ FAULT1/OVF/ UFS<br>127 : CNT/TRG<br>128 : MC0<br>129 : MC1 | **APB:**<br>CLKMSK[1]<br>MASK16 | PCHCTRL[32] | 38<br>STATUS1<br>PERID6 | 34 : OVF<br>35 : MC0<br>36 : MC1 | Generators:<br>**OVF:**<br>CHANNEL[71]<br>**TRG:**<br>CHANNEL[72]<br>**CNT:**<br>CHANNEL[73]<br>**MC0:**<br>CHANNEL[74]<br>**MC1:**<br>CHANNEL[75]<br>Users:<br>**EV0:**<br>USER[34],Resynch<br>**EV1:**<br>USER[35],Resynch<br>**MC0:**<br>USER[36],Resynch<br>**MC1:**<br>USER[37],Resynch | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 38.7. Register Summary

| Offset | Name | Bit Pos. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x00 | CTRLA | 7:0 | | | | | | | ENABLE | SWRST |
| | | 15:8 | | ALOCK | PRESCSYNC[1:0] | | RUNSTDBY | PRESCALER[2:0] | | |
| | | 23:16 | DMAOS | | | | | | | FCYCLE |
| | | 31:24 | CPTEN7 | CPTEN6 | 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 | | | | | | | | |
| | | 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 ... 0x17 | Reserved | | | | | | | | | |
| 0x18 | DRVCTRL | 7:0 | NRE | NRE | NRE | NRE | NRE | NRE | NRE | NRE |
| | | 15:8 | NRV | NRV | NRV | NRV | NRV | NRV | NRV | NRV |
| | | 23:16 | INVEN | INVEN | INVEN | INVEN | INVEN | INVEN | INVEN | INVEN |
| | | 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 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x24 | INTENCLR | 7:0 | | | | | ERR | CNT | TRG | OVF |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x28 | INTENSET | 7:0 | | | | | ERR | CNT | TRG | OVF |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x2C | INTFLAG | 7:0 | | | | | ERR | CNT | TRG | OVF |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | DFS | UFS | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x30 | STATUS | 7:0 | PERBUFV | | PATTBUFV | | DFS | UFS | IDX | STOP |
| | | 15:8 | FAULT1 | FAULT0 | FAULTB | FAULTA | FAULT1IN | FAULT0IN | FAULTBIN | FAULTAIN |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x34 | COUNT | 7:0 | COUNT[7:0] | | | | | | | |
| | | 15:8 | COUNT[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 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x38 ... 0x3B | Reserved | | | | | | | | | |
| 0x3C | WAVE | 7:0 | CIPEREN | RAMP[2:0] | | | | WAVEGEN[2:0] | | |
| | | 15:8 | | | | | CICCEN3 | CICCEN2 | CICCEN1 | CICCEN0 |
| | | 23:16 | | | | | | | POL1 | POL0 |
| | | 31:24 | | | | | | | | |
| 0x40 | PER | 7:0 | PER[7:0] | | | | | | | |
| | | 15:8 | PER[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x44 | CCy0 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x48 | CCy1 | 7:0 | CC[7:0] | | | | | | | |
| | | 15:8 | CC[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x4C ... 0x6B | Reserved | | | | | | | | | |
| 0x6C | PERBUF | 7:0 | PERBUF[7:0] | | | | | | | |
| | | 15:8 | PERBUF[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x70 | CCBUFy0 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |
| 0x74 | CCBUFy1 | 7:0 | CCBUF[7:0] | | | | | | | |
| | | 15:8 | CCBUF[15:8] | | | | | | | |
| | | 23:16 | | | | | | | | |
| | | 31:24 | | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.1. Control A

**Name:** CTRLA
**Offset:** 0x00
**Reset:** 0x00000000
**Property:** PAC Write-Protection

**Table 38-6.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | CPTEN7 | CPTEN6 | CPTEN5 | CPTEN4 | CPTEN3 | CPTEN2 | CPTEN1 | CPTEN0 |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | DMAOS | | | | | | | FCYCLE |
| Access | R/W | | | | | | | R/W |
| Reset | 0 | | | | | | | 0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|---|
| | | ALOCK | PRESCSYNC[1:0] | | RUNSTDBY | PRESCALER[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 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | ENABLE | SWRST |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 0 | 0 |

**Bits 24, 25, 26, 27, 28, 29, 30, 31 – CPTEN** Capture Channel x EnableThese 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.These bits are enable-protected. These bits are not synchronized.

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.
This bit is enable-protected.

**Bit 23 – DMAOS** DMA One-shot Trigger ModeThis 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 enable-protected. This bit is not synchronized.

**Bit 16 – FCYCLE** Full Cycle Enable

When this bit is set, TCC will wait for the end of the current cycle, to evaluate the stop condition.
This bit is enable-protected.

| Value | Description |
|---|---|
| 0 | The stop condition is evaluated immediately. |
| 1 | The stop condition is evaluated at the end of the cycle. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 14 – ALOCK** Auto Lock
This bit is enable-protected.

| 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. |

**Bits 13:12 – PRESCSYNC[1:0]** Prescaler and Counter Synchronization Selection
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 enable-protected.

| 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 enable-protected.

| 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 enable-protected.

| Value | Name | Description |
|---|---|---|
| 0x0 | DIV1 | Prescaler: GCLK_TCC |
| 0x1 | DIV2 | Prescaler: GCLK_TCC/2 |
| 0x2 | DIV4 | Prescaler: GCLK_TCC/4 |
| 0x3 | DIV8 | Prescaler: GCLK_TCC/8 |
| 0x4 | DIV16 | Prescaler: GCLK_TCC/16 |
| 0x5 | DIV64 | Prescaler: GCLK_TCC/64 |
| 0x6 | DIV256 | Prescaler: GCLK_TCC/256 |
| 0x7 | DIV1024 | Prescaler: GCLK_TCC/1024 |

**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.

| 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

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.
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.

### 38.7.2. 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.

**Table 38-7.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 read back zero. The commands are executed on the next prescaled GCLK_TCCx clock cycle.
Writing zero to this bit group has no effect.
Writing a '1' to any of these bits will clear the pending command.

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No action |
| 1 | RETRIGGER | Clear start, restart or retrigger |
| 2 | STOP | Force stop |
| 3 | UPDATE | Force update or double buffered registers |
| 4 | READSYNC | Force COUNT read synchronization |
| 5 | DMAOS | One-shot DMA trigger |

**Bits 4:3 – IDXCMD[1:0]** Ramp Index Command
These bits can be used to force cycle A (Ramp A) and cycle B (Ramp B) changes in all RAMP2x operations. 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 |
|---|---|---|
| 0 | DISABLE | Command disabled: Index toggles between cycles A and B |
| 1 | SET | Set index: cycle B will be forced in the next cycle |
| 2 | CLEAR | Clear index: cycle A will be forced in the next cycle |
| 3 | HOLD | Hold index: 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.
Reading this bit gives the following status:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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 CTRLBCLR.LUPD (CTRLBCLR<1>) 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.
Reading this bit gives the following status:

| Value | Description |
|---|---|
| 0 | The CCBUFy, PERBUF buffer register bitfields *are* copied into the corresponding CCy, PER registers and register bitfields on hardware update condition. |
| 1 | The CCBUFy, PERBUF buffer registers bitfields are *not* copied into the corresponding CCy, PER registers and register bitfields 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. Reading this bit gives the following status:

| 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.

### 38.7.3. 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.

**Table 38-8.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | CMD[2:0] | | | IDXCMD[1:0] | | ONESHOT | LUPD | DIR |
| Access | | | | | | | | |
| 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 valid value to this bit group will set the associated command.

| Value | Name | Description |
|---|---|---|
| 0 | NONE | No action |
| 1 | RETRIGGER | Clear start, restart or retrigger |
| 2 | STOP | Force stop |
| 3 | UPDATE | Force update or double buffered registers |
| 4 | READSYNC | Force COUNT read synchronization |
| 5 | 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 all RAMP2x operations. 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 |
|---|---|---|
| 0 | DISABLE | Command disabled: Index toggles between cycles A and B |
| 1 | SET | Set index: cycle B will be forced in the next cycle |
| 2 | CLEAR | Clear index: cycle A will be forced in the next cycle |
| 3 | HOLD | Hold index: 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.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will enable the one-shot operation.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

| 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 CTRLBSET.LUPD (CTRLBSET<1>) 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 CCBUFy, PERBUF buffer register bitfields *are* copied into the corresponding CCy, PER registers and register bitfields on hardware update condition. |
| 1 | The CCBUFy, PERBUF buffer registers bitfields are *not* copied into the corresponding CCy, PER registers and register bitfields 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. Reading this bit gives following status:

| 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.

### 38.7.4. Synchronization Busy

**Name:** SYNCBUSY
**Offset:** 0x08
**Reset:** 0x00000000
**Property:** -

**Table 38-9.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 7 – PER** Period 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 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** Pattern 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 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.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 3 – STATUS** Status 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 Busy
This bit is cleared when the synchronization of CTRLBSET/CTRLBCLR register between the clock domains is complete.
This bit is set when the synchronization of CTRLBSET/CTRLBCLR register between clock domains is started.

**Bit 1 – ENABLE** Enable 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 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.

### 38.7.5. Recoverable Fault A Configuration

**Name:** FCTRLA
**Offset:** 0x0C
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 38-10.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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]** Fault A Filter Value
These bits define the filter value applied on the Fault A (MCE0) event input line. Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed to the TCC. The value must be set to zero when this event is used as synchronous event.

**Bits 23:16 – BLANKVAL[7:0]** Fault A Blanking Time
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 (FCTRLA.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** Fault A Blanking 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]** Fault A Capture Action
These bits select the capture and Fault A interrupt/event conditions.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-11.** Fault A Capture Action

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Capture on valid recoverable Fault A is disabled |
| 0x1 | CAPT | On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each new captured value. |
| 0x2 | CAPTMIN | On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) , if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local minimum detection. |
| 0x3 | CAPTMAX | On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) , if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTA (INTFLAG<12>)flag rises on each local maximum detection. |
| 0x4 | LOCMIN | On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local minimum value detection. |
| 0x5 | LOCMAX | On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local maximum detection. |
| 0x6 | DERIV0 | On rising edge of a valid recoverable Fault A, capture counter value on channel selected by FCTRLA.CHSEL bits (FCTRLA <11:10>) . INTFLAG.FAULTA (INTFLAG<12>) flag rises on each local maximum or minimum detection. |
| 0x7 | CAPTMARK | Capture with Ramp index as MSB value. |

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | No capture |
| 1 | CAPT | Capture on fault |
| 2 | CAPTMIN | Minimum capture |
| 3 | CAPTMAX | Maximum capture |
| 4 | LOCMIN | Minimum local detection |
| 5 | LOCMAX | Maximum local detection |
| 6 | DERIV0 | Minimum and maximum local detection |
| 7 | CAPTMARK | Capture with ramp index as MSB value |

**Bits 11:10 – CHSEL[1:0]** Fault A Capture Channel

These bits select the channel for capture operation triggered by recoverable Fault A.

| Value | Name | Description |
|---|---|---|
| 0 | CC0 | Capture value stored in channel 0 |
| 1 | CC1 | Capture value stored in channel 1 |
| 2 | CC2 | Capture value stored in channel 2 |
| 3 | CC3 | Capture value stored in channel 3 |

**Bits 9:8 – HALT[1:0]** Fault A Halt Mode

These bits select the halt action for recoverable Fault A.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Halt action disabled |
| 1 | HW | Hardware halt action |
| 2 | SW | Software halt action |
| 3 | NR | Non-recoverable fault |

**Bit 7 – RESTART** Fault A Restart

Setting this bit enables restart action for Fault A.

| Value | Description |
|---|---|
| 0 | Fault A restart action is disabled. |
| 1 | Fault A restart action is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 6:5 – BLANK[1:0]** Fault A Blanking Mode
These bits, select the blanking start point for recoverable Fault A.

| Value | Name | Description |
|---|---|---|
| 0 | START | Blanking applied from start of the ramp |
| 1 | RISE | Blanking applied from rising edge of the output waveform |
| 2 | FALL | Blanking applied from falling edge of the output waveform |
| 3 | BOTH | Blanking applied from each toggle of the output waveform |

**Bit 4 – QUAL** Fault A Qualification
Setting this bit enables the recoverable Fault A input qualification.

| Value | Description |
|---|---|
| 0 | The recoverable Fault A input is not disabled on CMPy value condition. |
| 1 | The recoverable Fault A input is disabled when channel output is at inactive level (CMPy = 0). |

**Bit 3 – KEEP** Fault A Keeper
Setting this bit enables the Fault A keep action.

| Value | Description |
|---|---|
| 0 | The Fault A state is released as soon as the recoverable Fault A is released. |
| 1 | The Fault A state is released at the end of TCC cycle. |

**Bits 1:0 – SRC[1:0]** Fault A Source
These bits select the TCC Fault A mode.
Event system channel connected to Match or Capture Channel 0 (MCE0) event input, must be configured to route the event asynchronously, when used as a recoverable Fault A input.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Fault input disabled |
| 1 | ENABLE | MCEx (x=0,1) event input |
| 2 | INVERT | Inverted MCEx (x=0,1) event input |
| 3 | 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.

### 38.7.6. Recoverable Fault B Configuration

**Name:** FCTRLB
**Offset:** 0x10
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 38-12.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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]** Fault B Filter Value
These bits define the filter value applied on FAULT B (MCE1) event input line. An Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed. The value must be set to zero when this event is used as synchronous event.

**Bits 23:16 – BLANKVAL[7:0]** Fault B Blanking Time
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 (FCTRLB.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** Fault B Blanking 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]** Fault B Capture Action
These bits select the capture and Fault B interrupt/event conditions.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 38-13.** Fault B Capture Action

| Value | Name | Description |
|---|---|---|
| 0x0 | DISABLE | Capture on valid recoverable Fault B is disabled |
| 0x1 | CAPT | On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each new captured value. |
| 0x2 | CAPTMIN | On rising edge of a valid recoverable Fault B, capture counter value on channel selected byFCTRLB.CHSEL bits (FCTRLB <11:10>), if COUNT value is lower than the last stored capture value (CC). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local minimum detection. |
| 0x3 | CAPTMAX | On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>), if COUNT value is higher than the last stored capture value (CC). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local maximun detection. |
| 0x4 | LOCMIN | On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local minimum value detection. |
| 0x5 | LOCMAX | On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local maximun detection. |
| 0x6 | DERIV0 | On rising edge of a valid recoverable Fault B, capture counter value on channel selected by FCTRLB.CHSEL bits (FCTRLB <11:10>). INTFLAG.FAULTB (INTFLAG<13>) flag rises on each local maximun or minimum detection. |
| 0x7 | CAPTMARK | Capture with Ramp index as MSB value. |

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | No capture |
| 1 | CAPT | Capture on fault |
| 2 | CAPTMIN | Minimum capture |
| 3 | CAPTMAX | Maximum capture |
| 4 | LOCMIN | Minimum local detection |
| 5 | LOCMAX | Maximum local detection |
| 6 | DERIV0 | Minimum and maximum local detection |
| 7 | CAPTMARK | Capture with ramp index as MSB value |

**Bits 11:10 – CHSEL[1:0]** Fault B Capture Channel

These bits select the channel for capture operation triggered by recoverable Fault B.

| Value | Name | Description |
|---|---|---|
| 0 | CC0 | Capture value stored in channel 0 |
| 1 | CC1 | Capture value stored in channel 1 |
| 2 | CC2 | Capture value stored in channel 2 |
| 3 | CC3 | Capture value stored in channel 3 |

**Bits 9:8 – HALT[1:0]** Fault B Halt Mode

These bits select the halt action for recoverable Fault B.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Halt action disabled |
| 1 | HW | Hardware halt action |
| 2 | SW | Software halt action |
| 3 | NR | Non-recoverable fault |

**Bit 7 – RESTART** Fault B Restart

Setting this bit enables restart action for Fault B.

| Value | Description |
|---|---|
| 0 | Fault B restart action is disabled. |
| 1 | Fault B restart action is enabled. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bits 6:5 – BLANK[1:0]** Fault B Blanking Mode
These bits, select the blanking start point for recoverable Fault B.

| Value | Name | Description |
|---|---|---|
| 0 | START | Blanking applied from start of the ramp |
| 1 | RISE | Blanking applied from rising edge of the output waveform |
| 2 | FALL | Blanking applied from falling edge of the output waveform |
| 3 | BOTH | Blanking applied from each toggle of the output waveform |

**Bit 4 – QUAL** Fault B Qualification
Setting this bit enables the recoverable Fault B input qualification.

| Value | Description |
|---|---|
| 0 | The recoverable Fault B input is not disabled on CMPy value condition. |
| 1 | The recoverable Fault B input is disabled when output signal is at inactive level (CMPy = 0). |

**Bit 3 – KEEP** Fault B Keeper
Setting this bit enables the Fault B keep action.

| Value | Description |
|---|---|
| 0 | The Fault B state is released as soon as the recoverable Fault B is released. |
| 1 | The Fault B state is released at the end of TCC cycle. |

**Bits 1:0 – SRC[1:0]** Fault B Source
These bits select the TCC Fault B mode.
Event system channel connected to Match or Capture Channel 1 (MCE1) event input, must be configured to route the event asynchronously, when used as a recoverable Fault B input.

| Value | Name | Description |
|---|---|---|
| 0 | DISABLE | Fault input disabled |
| 1 | ENABLE | MCEx (x=0,1) event input |
| 2 | INVERT | Inverted MCEx (x=0,1) event input |
| 3 | 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.

### 38.7.7. Driver Control

**Name:** DRVCTRL
**Offset:** 0x18
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 38-14.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | INVEN | INVEN | INVEN | INVEN | INVEN | INVEN | INVEN | INVEN |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | NRV | NRV | NRV | NRV | NRV | NRV | NRV | NRV |
| Access | R/W | 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 |
|---|---|---|---|---|---|---|---|---|
| | NRE | NRE | NRE | NRE | NRE | NRE | NRE | NRE |
| Access | R/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 Fault 1 (TCE1) event input line. An Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed. When value is 0, Fault1 acts asynchronously on PWM outputs.

**Bits 27:24 – FILTERVAL0[3:0]** Non-Recoverable Fault Input 0 Filter Value
These bits define the filter value applied on Fault 0 (TCE0) event input line. An Input signal with pulse width shorter than (FILTERVAL) * GCLK_TCCx is filtered. An Input signal with pulse width larger than (FILTERVAL+1) * GCLK_TCCx is passed. When value is 0, Fault 0 acts asynchronously on PWM outputs.

**Bits 16, 17, 18, 19, 20, 21, 22, 23 – INVEN** Output Waveform 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** Non-Recoverable State x Output Value
These bits define the value of the enabled override outputs, under non-recoverable fault condition.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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.

### 38.7.8. Debug Control

**Name:** DBGCTRL
**Offset:** 0x1E
**Reset:** 0x00
**Property:** PAC Write-Protection

**Table 38-15.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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.

| Value | Description |
|---|---|
| 0 | No faults are generated when TCC is halted in Debug mode. |
| 1 | A non recoverable fault is generated and INTFLAG.DFS (INTFLAG<11>) flag is set when TCC is halted in Debug mode. |

**Bit 0 – DBGRUN** Debug Running Mode
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.

### 38.7.9. Event Control

**Name:** EVCTRL
**Offset:** 0x20
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Enable-Protected

**Table 38-16.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | 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 14, 15 – TCEI** Timer/counter Event x Input 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** Inverted Event x Input Enable
This bit inverts the event x input.

**Bit 10 – CNTEO** Timer/counter Output Event 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 on EVCTRL.CNTSEL(EVCTRL<7:6>) 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 depending on EVCTRL.CNTSEL(EVCTRL<7:6>) value. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Bit 9 – TRGEO** Retrigger Output Event 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 Output Event 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 Output Event Mode

These bits define on which part of the counter cycle the counter event output is generated.

| Value | Name | Description |
|---|---|---|
| 0 | START | An interrupt/event is generated when a new counter cycle starts |
| 1 | END | An interrupt/event is generated when a counter cycle ends |
| 2 | BETWEEN | An interrupt/event is generated when a counter cycle ends, except for the first and last cycles |
| 3 | BOUNDARY | An interrupt/event is generated when a new counter cycle starts or a counter cycle ends |

**Bits 5:3 – EVACT1[2:0]** Timer/counter Input Event1 Action

These bits define the action the TCC will perform on TCE1 event input.

| Value | Name | Description |
|---|---|---|
| 0 | OFF | Event action disabled |
| 1 | RETRIGGER | Re-trigger counter on event |
| 2 | DIR | Direction control |
| 3 | STOP | Stop counter on event |
| 4 | DEC | Decrement counter on event |
| 5 | PPW | Period capture value in CC0 register, pulse width capture value in CC1 register |
| 6 | PWP | Period capture value in CC1 register, pulse width capture value in CC0 register |
| 7 | FAULT | Non-recoverable fault |

**Bits 2:0 – EVACT0[2:0]** Timer/counter Input Event0 Action

These bits define the action the TCC will perform on TCE0 event input 0.

| Value | Name | Description |
|---|---|---|
| 0 | OFF | Event action disabled |
| 1 | RETRIGGER | Start, restart or re-trigger counter on event |
| 2 | COUNTEV | Count on event |
| 3 | START | Start counter on event |
| 4 | INC | Increment counter on event |
| 5 | COUNT | Count on active state of asynchronous event |
| 6 | STAMP | Stamp capture |
| 7 | FAULT | Non-recoverable fault |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.10. 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.

**Table 38-17.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 15 – FAULT1** Non-Recoverable Fault 1 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault 1 Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault 1 interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault 1 interrupt is disabled. |
| 1 | The Non-Recoverable Fault 1 interrupt is enabled. |

**Bit 14 – FAULT0** Non-Recoverable Fault 0 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault 0 Interrupt Disable/Enable bit, which disables the Non-Recoverable Fault 0 interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault 0 interrupt is disabled. |
| 1 | The Non-Recoverable Fault 0 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.

| 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 Error 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. |

**Bit 1 – TRG** Retrigger 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 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.

### 38.7.11. 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.

**Table 38-18.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 15 – FAULT1** Non-Recoverable Fault 1 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will set the Non-Recoverable Fault 1 Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault 1 interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault 1 interrupt is disabled. |
| 1 | The Non-Recoverable Fault 1 interrupt is enabled. |

**Bit 14 – FAULT0** Non-Recoverable Fault 0 Interrupt Enable
Writing a '0' to this bit has no effect.
Writing a '1' to this bit will clear the Non-Recoverable Fault 0 Interrupt Disable/Enable bit, which enables the Non-Recoverable Fault 0 interrupt.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault 0 interrupt is disabled. |
| 1 | The Non-Recoverable Fault 0 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.

| 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 Error 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. |

**Bit 1 – TRG** Retrigger 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 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.

### 38.7.12. Interrupt Flag Status and Clear

**Name:** INTFLAG
**Offset:** 0x2C
**Reset:** 0x00000000
**Property:** -

**Note:** Interrupt flags must be cleared and then read back to confirm they are cleared before exiting the ISR to avoid double interrupts.

**Table 38-19.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | 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 |

**Bit 15 – FAULT1** Non-Recoverable Fault 1
This flag is set and resynchronized on the APB clock after a Non-Recoverable Fault 1 occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault 1 interrupt flag.

**Bit 14 – FAULT0** Non-Recoverable Fault 0
This flag is set and resynchronized on the APB clock after a Non-Recoverable Fault 0 occurs.
Writing a '0' to this bit has no effect.
Writing a '1' to this bit clears the Non-Recoverable Fault 0 interrupt flag.

| Value | Description |
|---|---|
| 0 | The Non-Recoverable Fault y interrupt is disabled. |
| 1 | The Non-Recoverable Fault y interrupt is enabled. |

**Bit 13 – FAULTB** Recoverable Fault B
This flag is set and resynchronized on the APB clock 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
This flag is set and resynchronized on the APB clock 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
This flag is set and resynchronized on the APB clock 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(CTRLBSET<1>)).
Writing a zero to this bit has no effect.
Writing a one to this bit clears the Non-Recoverable Update Fault interrupt flag.

**Bit 3 – ERR** Error
This flag is set if a new capture occurs on a channel when the corresponding Match or Capture Channel y 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
This flag is set and resynchronized on the APB clock 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
This flag is set and resynchronized on the APB clock 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
This flag is set and resynchronized on the APB clock 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.

### 38.7.13. Status

**Name:** STATUS
**Offset:** 0x30
**Reset:** 0x00000001
**Property:** -

**Table 38-20.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | FAULT1 | FAULT0 | FAULTB | FAULTA | FAULT1IN | FAULT0IN | FAULTBIN | FAULTAIN |
| Access | - | - | 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 | | DFS | UFS | IDX | STOP |
| Access | R/W | | R/W | | R/W | R/W | R | R |
| Reset | 0 | | 0 | | 0 | 0 | 0 | 1 |

**Bit 15 – FAULT1** Non-Recoverable Fault 1 State
This bit is set by hardware as soon as non-recoverable Fault 1 condition occurs.
This bit is cleared by writing a one to this bit and when the corresponding STATUS.FAULT1IN(STATUS<11>) status bit is low.
Once this bit is cleared, the TCC will restart from the last COUNT value. To restart the TCC from BOTTOM, the TCC restart (RETRIGGER) command (CTRLBSET.CMD (CTRLBSET<7:5>)=1) must be executed before clearing the FAULT1 STATE bit. For further details on TCC commands, refer to the available commands description in the CTRLBSET register.

**Bit 14 – FAULT0** Non-Recoverable Fault 0 State
This bit is set by hardware as soon as non-recoverable Fault 0 condition occurs.
This bit is cleared by writing a one to this bit and when the corresponding STATUS.FAULT0IN(STATUS<10>) status bit is low.
Once this bit is cleared, the TCC will restart from the last COUNT value. To restart the TCC from BOTTOM, the TCC restart (RETRIGGER) command (CTRLBSET.CMD (CTRLBSET<7:5>)=1) must be executed before clearing the FAULT0 STATE bit. For further details on TCC commands, refer to the available commands description in the CTRLBSET register.

**Bit 13 – FAULTB** Recoverable Fault B State
This bit is set by hardware as soon as recoverable Fault B condition occurs.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

This bit can be clear by the 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 (FCTRLB.HALT (FCTRLB<9:8>) =SW), clearing this bit will release the timer/counter.

**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 the 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 (FCTRLA.HALT (FCTRLB<9:8>) =SW), clearing this bit will release the timer/counter.

**Bit 11 – FAULT1IN** Non-Recoverable Fault1 Input
This bit is set while an active Non-Recoverable Fault 1 input is present.

**Bit 10 – FAULT0IN** Non-Recoverable Fault0 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 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 3 – DFS** Non-Recoverable 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
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 the "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.

### 38.7.14. Count

**Name:** COUNT
**Offset:** 0x34
**Reset:** 0x00000000
**Property:** PAC Write-Protection, Write-Synchronized, Read-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).

**Table 38-21.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 hold the value of the Counter register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.15. Waveform Control

**Name:** WAVE
**Offset:** 0x3C
**Reset:** 0x00000000
**Property:** Write-Synchronized

**Table 38-22.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Access | | | | | | | | |
| Reset | | | | | | | | |

| Bit | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | POL1 | POL0 |
| Access | | | | | | | R/W | R/W |
| Reset | | | | | | | 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[2:0] | | | | WAVEGEN[2:0] | | |
| Access | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | | 0 | 0 | 0 |

**Bits 16, 17 – POLx** Channel Polarity x [x = 3..0]
Setting these bits enables the output polarity in single-slope and dual-slope PWM operations.
**Note:** These bits are write-synchronized. SYNCBUSY.WAVE must be checked to ensure that WAVE.POLx synchronization is complete.

| 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 – CICCENx** Circular CC Enable x [x = 3..0]
Setting this bits enables the compare circular buffer option on channel. When the bit is set, CCx register value is copied-back into the CCx register on UPDATE condition.
**Note:** These bits are write-synchronized. SYNCBUSY.WAVE must be checked to ensure that WAVE.CICCENx synchronization is complete.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**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.

**Note:** This bit is write-synchronized. SYNCBUSY.WAVE must be checked to ensure that WAVE.CIPEREN synchronization is complete.

**Bits 6:4 – RAMP[2:0]** Ramp Mode

These bits select Ramp operation (RAMP). These bits are not synchronized.

| Value | Name | Description |
|---|---|---|
| 0 | RAMP1 | RAMP1 operation |
| 1 | RAMP2A | Alternative RAMP2 operation |
| 2 | RAMP2 | RAMP2 operation |
| 3 | RAMP2C | Critical RAMP2 operation |
| 4 | RAMP2CS | Critical Swapped RAMP2 operation |

**Bits 2:0 – WAVEGEN[2:0]** Waveform Generation

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** | OVF Interrupt Flag/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 | DPWM | Dual Compare PWM | PER | TOP/ZERO | Set/Clear | Clear | - | Zero |
| 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 | – |

| Value | Name | Description |
|---|---|---|
| 0 | NFRQ | Normal frequency |
| 1 | MFRQ | Match frequency |
| 2 | NPWM | Normal PWM |
| 3 | DPWM | Dual compare PWM |
| 4 | DSCRITICAL | Dual-slope critical |
| 5 | DSBOTTOM | Dual-slope with interrupt/event condition when COUNT reaches ZERO |
| 6 | DSBOTH | Dual-slope with interrupt/event condition when COUNT reaches ZERO or TOP |
| 7 | DSTOP | Dual-slope with interrupt/event condition when COUNT reaches TOP |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.16. Period

**Name:** PER
**Offset:** 0x40
**Reset:** 0xFFFFFFFF
**Property:** Write-Synchronized

**Table 38-23.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PER[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 |
|---|---|---|---|---|---|---|---|---|
| | PER[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 – PER[15:0]** Period Value


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.17. Compare and Capture

**Name:** CCy
**Offset:** 0x44 + y*0x04 [y=0..1]
**Reset:** 0x00000000
**Property:** Write-Synchronized, Read-Synchronized

The CCy register represents the 16-bit value. 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.

CCy register is updated with the buffer value from their corresponding CCBUFy register when an UPDATE condition occurs.

In addition, in match frequency operation, the CC0 register controls the counter period.

**Table 38-24.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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/Capture Value
These bits hold the value of the Channel y compare/capture register.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 38.7.18. Period Buffer

**Name:** PERBUF
**Offset:** 0x6C
**Reset:** 0xFFFFFFFF
**Property:** Write-Synchronized, Read-Synchronized

**Table 38-25.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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 |
|---|---|---|---|---|---|---|---|---|
| | PERBUF[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 |
|---|---|---|---|---|---|---|---|---|
| | PERBUF[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 – PERBUF[15:0]** 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.

### 38.7.19. Compare and Capture Buffer

**Name:** CCBUFy
**Offset:** 0x70 + y*0x04 [y=0..1]
**Reset:** 0x00000000
**Property:** Write-Synchronized, Read-Synchronized

CCBUFy is copied into CCy at TCC update time.

**Table 38-26.** Register Bit Attribute Legend

| Symbol | Description | Symbol | Description | Symbol | Description |
|---|---|---|---|---|---|
| **R** | Readable bit | **HC** | Cleared by Hardware | (Grey cell) | Unimplemented |
| **W** | Writable bit | **HS** | Set by Hardware | **X** | Bit is unknown at Reset |
| **K** | Write to clear | **S** | Software settable bit | — | — |

| 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/Capture Buffer Value
These bits hold the value of the Channel y Compare/Capture Buffer Value register. The register serves as the buffer for the associated compare or capture registers (CCy). Accessing this register using the CPU or DMA will affect the corresponding STATUS.CCBUFVy status bit.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 39. Peripheral Touch Controller (PTC)

## 39.1. Overview

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on 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/0 pins in the device. The PTC supports self-capacitance and mutual-capacitance sensors.

In 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 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-line and Y-line are depending on both package type and device configuration. Refer to the *Configuration Summary and I/0 Multiplexing* table for additional information.

**Important:** If AVDD < 2.5v, users must enable analog charge pumps in the SUPC.VREGCTRL.CPEN[1:0].

## 39.2. Features

The following are key features of the Peripheral Touch Controller module:

- Low-power, high-sensitivity, environmentally robust capacitive touch buttons, sliders, wheels and proximity sensing.
- Supports wake-up on touch from Standby Sleep mode
- Supports mutual-capacitance and self-capacitance sensing
  - Up to 1024 buttons in mutual-capacitance mode.
  - Up to 64 buttons in self-capacitance mode.
  - 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 VDD range
  - Auto calibration and recalibration of sensors
- Single-shot charge measurement
- Hardware noise filtering and noise signal desynchronization for high conducted immunity
- Polarity control, allowing Parallel Acquisition (through the QTouch Library) individually controls the polarity of each line
- Driven Shield+ for better noise immunity and moisture tolerance
  - Any PTC X/Y line can be used for the driven shield
  - All enabled sensors will be driven at the same potential as the sensor scanned
- 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


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 39.3. Block Diagram

**Figure 39-1.** PTC Block Diagram Mutual Capacitance

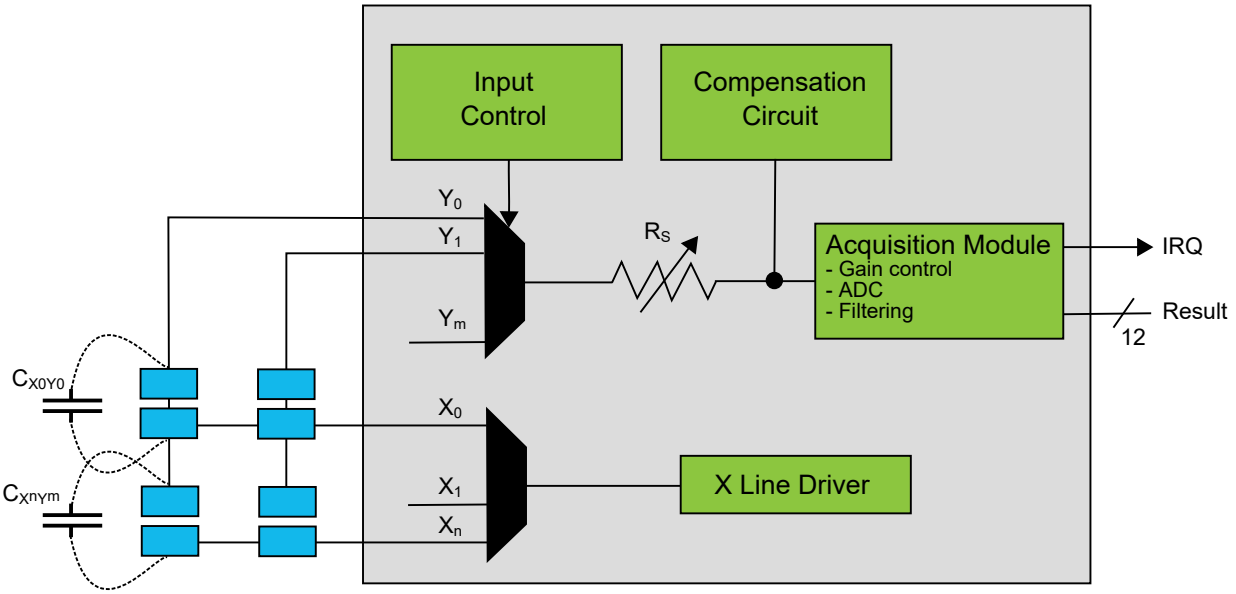
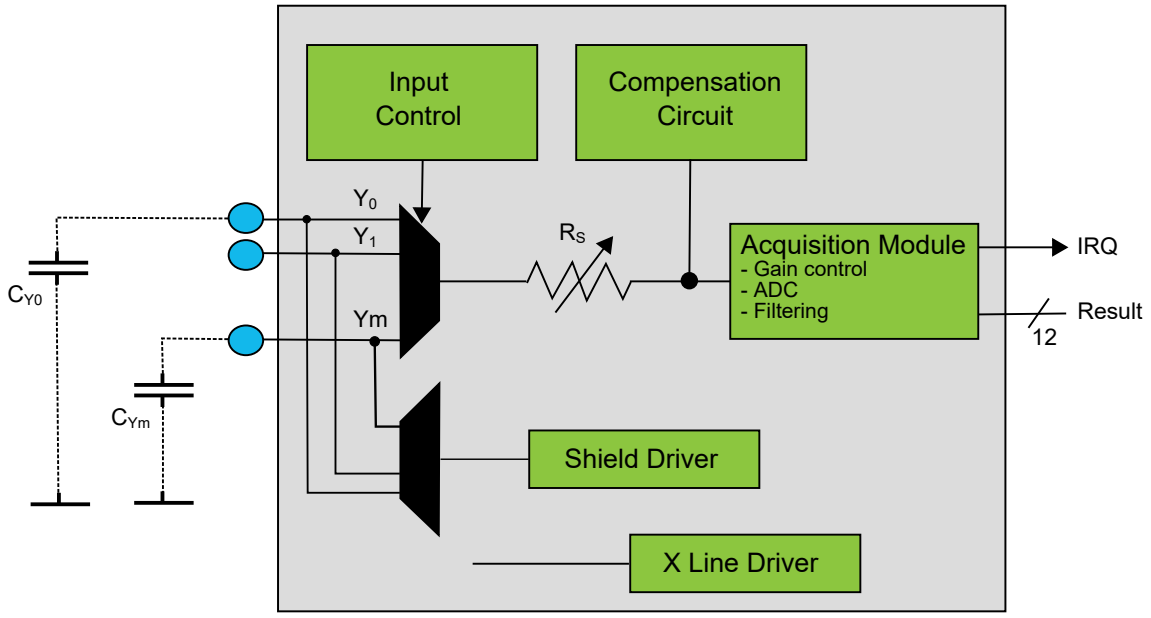



**Figure 39-2.** PTC Block Diagram Self Capacitance




## 39.4. Signal Description

**Table 39-1.** Signal Description for PTC

| Name | Type | Description |
|---|---|---|
| Y[m:0] | Analog | Y-line (Input/0utput) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 39-1.** Signal Description for PTC (continued)

| Name | Type | Description |
|---|---|---|
| X[n:0] | Digital | X-line (0utput) |

**Note:** The number of X-lines and Y-lines are device dependent. Refer to the Configuration Summary and Signal Description for additional information.

Refer to the Pinout section for details on the pin mapping for this peripheral. One signal can be mapped on several pins.

## 39.5. Peripheral Dependencies

In order to use this peripheral, configure the other components of the system as described in the following sections.

### 39.5.1. I/O Lines

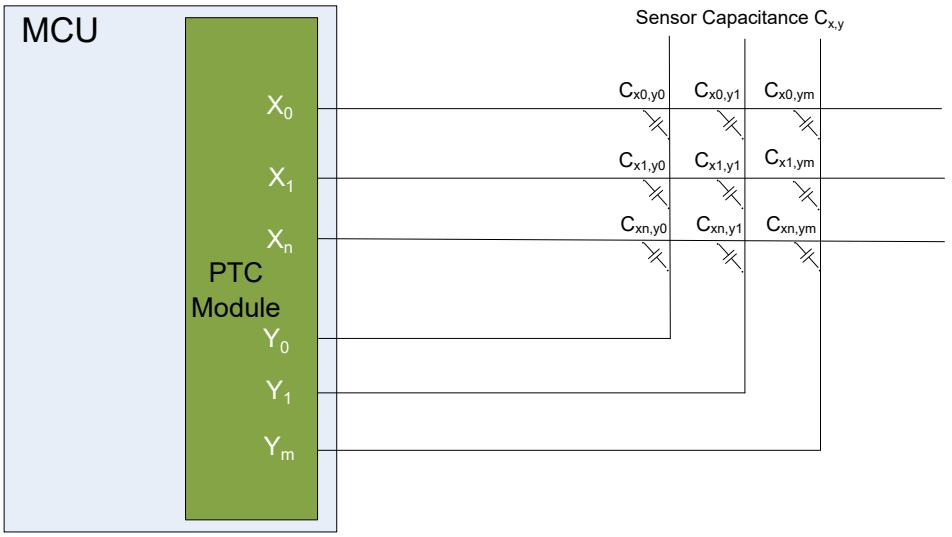
The I/0 lines used for analog X-lines and Y-lines must be connected to external capacitive touch sensor electrodes. External components are not required for normal operation. However, to improve the EMC performance, a series resistor of 1kΩ or more can be used on X-lines and Y-lines.

**Important:** Serial resistance are only applied on Y-lines. Each line might be either X or Y lines depending on the configuration (self vs. mutual capacitance), do not add serial resistor on X side in mutual capacitance operation.

#### 39.5.1.1. Mutual-Capacitance Sensor Arrangement

A mutual-capacitance sensor is formed between two I/0 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 39-3.** Mutual Capacitance Sensor Arrangement



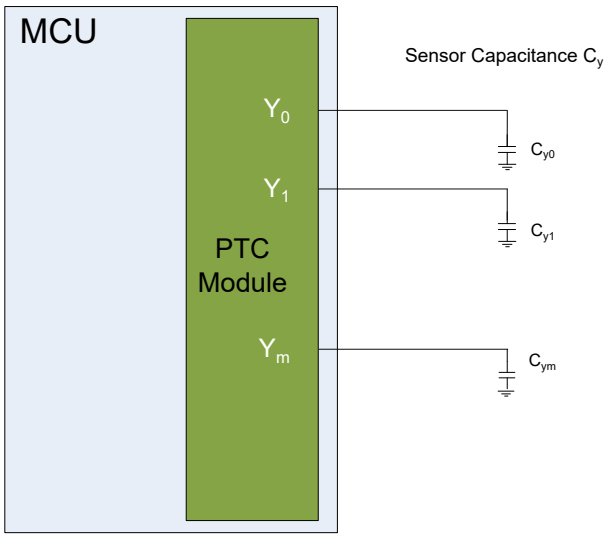


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

#### 39.5.1.2. 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.

**Figure 39-4.** Self-Capacitance Sensor Arrangement




For more information about designing the touch sensor, refer to Buttons, Sliders and Wheels Touch Sensor Design Guide.

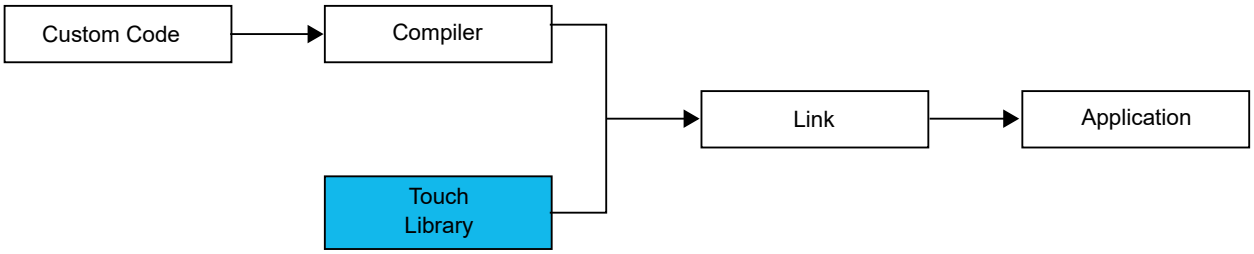
### 39.5.2. Clocks

The PTC is clocked by the GCLK_PTC clock. The PTC operates from an asynchronous clock source and the operation is independent of the main system clock and its derivative clocks, such as the peripheral bus clock (CLK_PTC_APB). A number of clock sources can be selected as the source for the asynchronous GCLK_PTC. The clock source is selected by configuring the Generic Clock Selection ID in the Generic Clock Control register. For more information about selecting the clock sources, refer to *GCLK - Generic Clock Controller* chapter.

## 39.6. Functional Description

In order to access the PTC, the user must use the Touch Configurator to configure and link the Touch Library firmware with the application software. Touch Library can be used to implement buttons, sliders, wheels in a variety of combinations on a single interface.

**Figure 39-5.** Touch Library Usage





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

For more information about Touch Library, refer to the Touch Library Peripheral Touch Controller User Guide.

### 39.6.1. Principle of Operation

The Peripheral Touch Controller (PTC) acquires signals in order to detect touch on 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-capacitance and mutual-capacitance sensors.

In 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 self-capacitance mode, the PTC requires only one pin (Y-line) for each touch sensor.

### 39.6.2. Initialization

The I/O Port pins that are chosen for touch sensing channels by the PTC module must be set to analog input/output mode using the Port control signals. GCLK_PTC must be set up to provide a clock for the module. The clock pre-scaler must also be set by writing a suitable value to CTLRA.PRSC register. The chosen X-channel and Y-channel set must be enabled for PTC use by writing the desired configuration into the PINEN control registers. The desired touch sensing mode is selected by writing the CTSM(2:0) bits in the CTRLB control register.

For operating the PTC, it must be enabled by setting the ENABLE bit in the CTRLA control register.

The external touch sensor node is selected by XSEL, and YSEL bits XSEL and YSEL registers, respectively. The Compensation capacitor value must be also initialized, by writing the settings to CCC in the CTRLC register. The Integration capacitor value must be likewise initialized by writing the settings to CI in the CTRLC register. The number of accumulations of the conversion results can be specified by writing DAN(2:0) bits in the CTRLD control register.

The module is disabled by writing a zero to the ENABLE bit in the CTRLA register.

### 39.6.3. Basic Operations

#### 39.6.3.1. Initialization

The following registers are enable-protected, meaning that they can only be written when the PTC is disabled (CTRLA.ENABLE = 0):

- The Event Control register (EVCTRL)
- The Pin Enable register (PINEN)
- The Calibration register (CALIB)

Enable-protection is denoted by the *Enable-Protected* property in the register description.

#### 39.6.3.2. Enabling, Disabling and Resetting

The PTC is enabled by writing a '1' to the Enable bit in the Control A register (CTRLA.ENABLE). The PTC is disabled by writing CTRLA.ENABLE = 0.

To go to the Active mode from the Off mode, the application must wait for the bias generators to stabilize, this wait is specified in powers of 2 by the Exponent bit field CTRLA.WKUPEXP[3:0]. The bit field CTRLA.WKUPEXP[3:0] generates the full count 2WKUPEXP, (where WKUPEXP is mapped internally to be always 4 or greater), which controls how many PTC core clocks are needed in the PTC core to fully power-up the analog bias circuitry. This delay is enforced every time CTRLA.ENABLE has a low to high transition. During this delay, the triggers are blocked from starting conversions, because the analog core is not ready yet. At the end of the delay, the corresponding status bit ACRRDY is set in the INTFLAG register for the user to poll and determine when the PTC Core is ready for operation and the triggers are allowed to start conversions. Because the delays are significant,


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

the ACRRDY status bits can be enabled as part of an group interrupt request. If the device entering in Standby Sleep mode during this delay without conversion ongoing, the wake-up time is hold.

The PTC is reset by writing a '1' to the Software Reset bit in the Control A register (CTRLA.SWRST). All registers in the PTC, except DBGCTRL, will be reset to their initial state, and the PTC will be disabled. Refer to CTRLA for additional information.

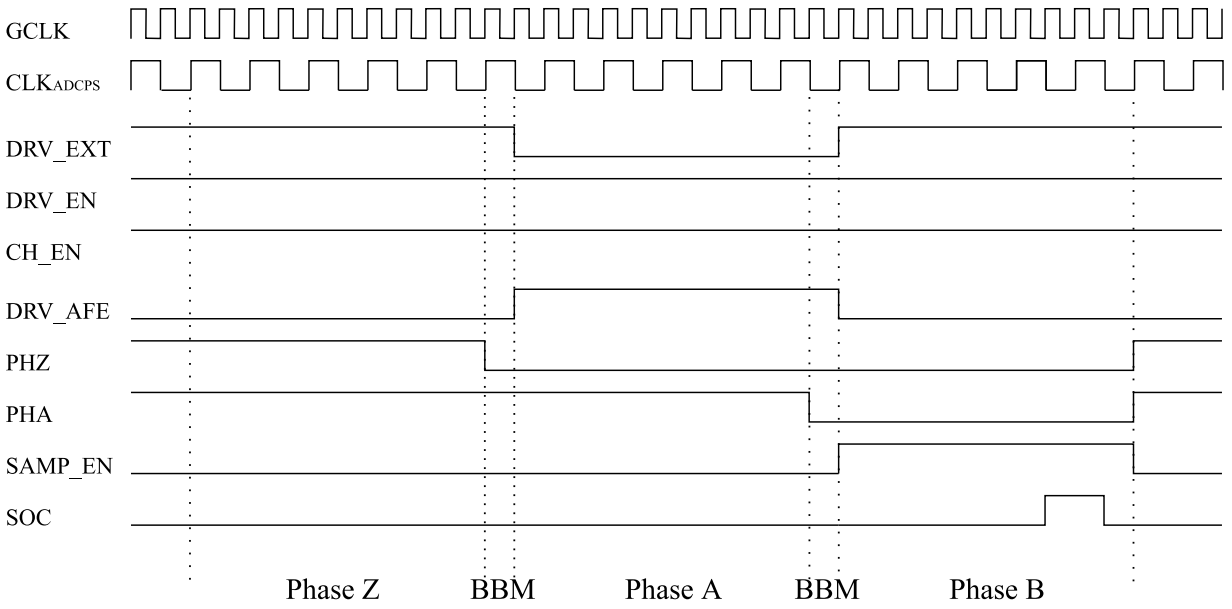
### 39.6.3.3. Mutual Capacitance Mode

The I/O Port pins that are chosen for touch sensing channels must be set to analog input/output mode using the Port control signals. The chosen X-channel and Y-channel set must be enabled for PTC use by writing the desired configuration into I/O Port Control registers. The desired touch sensing mode is selected by writing the CTSM(2:0) bits in the CTRLB control register. The compensation and the integration capacitor value must be also initialized by writing the settings to CTRLC. In all touch sensing mode, the timing of the various phases of the capacitive sensing is handling by the CSD bits in the CTRLC register.

**Table 39-2.** Timing in Mutual Cap mode

| | CSD | Phase Z | Phase A | Phase B | Min. Phase Time | Min. Conversion time |
|---|---|---|---|---|---|---|
| Differential Integration | 0 | NA | NA | NA | NA | NA |
| | 1 | 10 | CSD+1 | CSD+1 | 200 ns | 1 us |
| | 2 | 8 | CSD+1 | CSD+1 | 200 ns | 1 us |
| | 3 | 6 | CSD+1 | CSD+1 | 200 ns | 700 ns |
| | 4...255 | CSD+1 | CSD+1 | CSD+1 | 200 ns | 600 ns |

**Figure 39-6.** Differential Intergration of mutual cap sensor, normal polarity (CSD = 4)




The Peripheral Touch Controller is enabled by writing a one to the ENABLE bit in the Control A register (CTRLA.ENABLE).

The Peripheral Touch Controller is disabled by writing a zero to CTRLA.ENABLE.

After initialization, the sensor data conversion process is started by writing a one to the STCONV bit in the SWTRIG register. Once the specified number of charge integrations, ADC conversions and accumulations (1 to 64) are complete, EOC flag is set in the INTFLAG register


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

More advanced features include selectable channel change delay, when switching the X and Y multiplexers to a new channel. The selection is made by writing into the CCDS bits in the CTRLB register.

The touch sensor sampling instant can also be modified by writing a suitable code in the SDS bits in the CTRLB register. Furthermore, the sampling instant can be varied automatically by writing a one to the ASDV bit in the CTRLB register.

#### 39.6.3.4. Self Capacitance Mode

The I/O Port pins that are chosen for touch sensing channels must be set to analog input/output mode using the Port control signals. The chosen Y-channel set must be enabled for PTC use by writing the desired configuration into the PINEN control registers. The desired touch sensing mode is selected by writing the CTSM(3:0) bits in the CTRLC control register. The compensation and integration capacitor value must also be initialized, by writing the settings to the CTRLD register. In all touch sensing mode, the timing of the various phases of the capacitive sensing is handling by the CSD bits in the CTRLD, see the table below:

**Table 39-3.** Timing in Self Cap mode

| | CSD | IA | CB | IB | Min. Phase time | Min. conversion time |
|---|---|---|---|---|---|---|
| Differential Integration | 0 | Not Allowed | | | | |
| | 1 | CSD+1 | 5 | CSD+1 | 200 ns | 1.4 us |
| | 2 | CSD+1 | 4 | CSD+1 | 200 ns | 950 ns |
| | 3....255 | CSD+1 | CSD+1 | CSD+1 | 200 ns | 800 ns |

The Peripheral Touch Controller is enabled by writing a one to the ENABLE bit in the Control A register (CTRLA.ENABLE).

The Peripheral Touch Controller is disabled by writing a zero to CTRLA.ENABLE.

After initialization, the sensor data conversion process is started by writing a one to the STCONV bit in the SWTRIG register. Once the specified number of charge integrations, ADC conversions, and accumulations (1 to 64) are complete, the EOC flag is set in the INTFLAG register

More advanced features include selectable channel change delay, when switching the X and Y multiplexers to a new channel. The selection is made by writing into the CCDS bits in the CTRLB register.

The touch sensor sampling instant can be also modified by writing a suitable code in the SDS bits in the CTRLB register. Furthermore, the sampling instant can be varied automatically by writing a one to the ASDV bit in the CTRLB register.

### 39.6.4. Additional Features

#### 39.6.4.1. Digital Accumulation

**Digital Accumulation**

The result from multiple consecutive conversions can be accumulated. The number of samples to be accumulated is specified by the Digital Accumulation Number field in the CTRLD register (CTRLD.DAN). The accumulation result is available on the RES register. see table below for more details.

**Table 39-4.** Accumulation

| Number of Accumulated Samples | ACCCTRL. DAN | Final Result Precision |
|---|---|---|
| 1 | 0x0 | 12 bits |
| 2 | 0x1 | 13 bits |
| 4 | 0x2 | 14 bits |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 39-4.** Accumulation (continued)

| Number of Accumulated Samples | ACCCTRL. DAN | Final Result Precision |
|---|---|---|
| 8 | 0x3 | 15 bits |
| 16 | 0x4 | 16 bits |
| 32 | 0x5 | 16 bits (automatic division by 2) |
| 64 | 0x6 | 16 bits (automatic division by 4) |
| Reserved | 0x7 | |

**Analog Accumulation**

The analog accumulation is multiple sample acquisition and only one ADC conversion.

For mutual-cap mode, the Phase Z is performed only one time at the beginning of the acquisition, the Phase A and B are performed several times depending of CTRLD.AAN bits. The ADC conversion are done at the end of this loop and only one time. Therefore, the result is available in RES register on 12-bits.

For Self-cap mode, All phases are include in the acquisition loop, the loop number are programmed in CTRLD.AAN bits. The ADC conversion are done one time at the end of the loops. The result is available in RES register on 12-bits.

### 39.6.4.2. Window Monitor

The window monitor feature allows the conversion result in the RES register to be compared to predefined threshold values. The window mode is selected by setting the Window Monitor Mode bits in the Window Control register (CTRLD.WINCM). Threshold values must be written in the Window Monitor Low Threshold register (WINLT) and Window Monitor High Threshold register (WINHT).

When the Window Single Sample (CTRLD.WINSS) bit is written to '1', the window comparator is working on each sample instead of accumulated value. The number of samples matching with window comparator is available on Window Comparator Counter bits (STATUS.WCC).

The INTFLAG.WCOMP interrupt flag will be set if the conversion result matches the window monitor condition or if Window Comparator Counter is not zero in case of accumulation with Window Single Sample bit set.

**Note:** The CTRLD.WINSS bit works only with digital accumulation.

### 39.6.4.3. Driven Shield

Driven shield is the capability to drive an output line in synchronization with TOUCH input. This will provide better noise immunity from noise sources beneath the sensor, and improved sensitivity of the touch signal. Driven shield will only be used in self-cap mode.

### 39.6.4.4. Double Buffering

The following registers are double buffered:

- Control B (CTRLB)
- Control C (CTRLC)
- Control D (CTRLD)
- Window Monitor Low Threshold (WINLT)
- Window Monitor High Threshold (WINHT)
- X-line Selection (XSEL)
- Y-line Selection (YSEL)
- Polarity Control (POLCTRL)

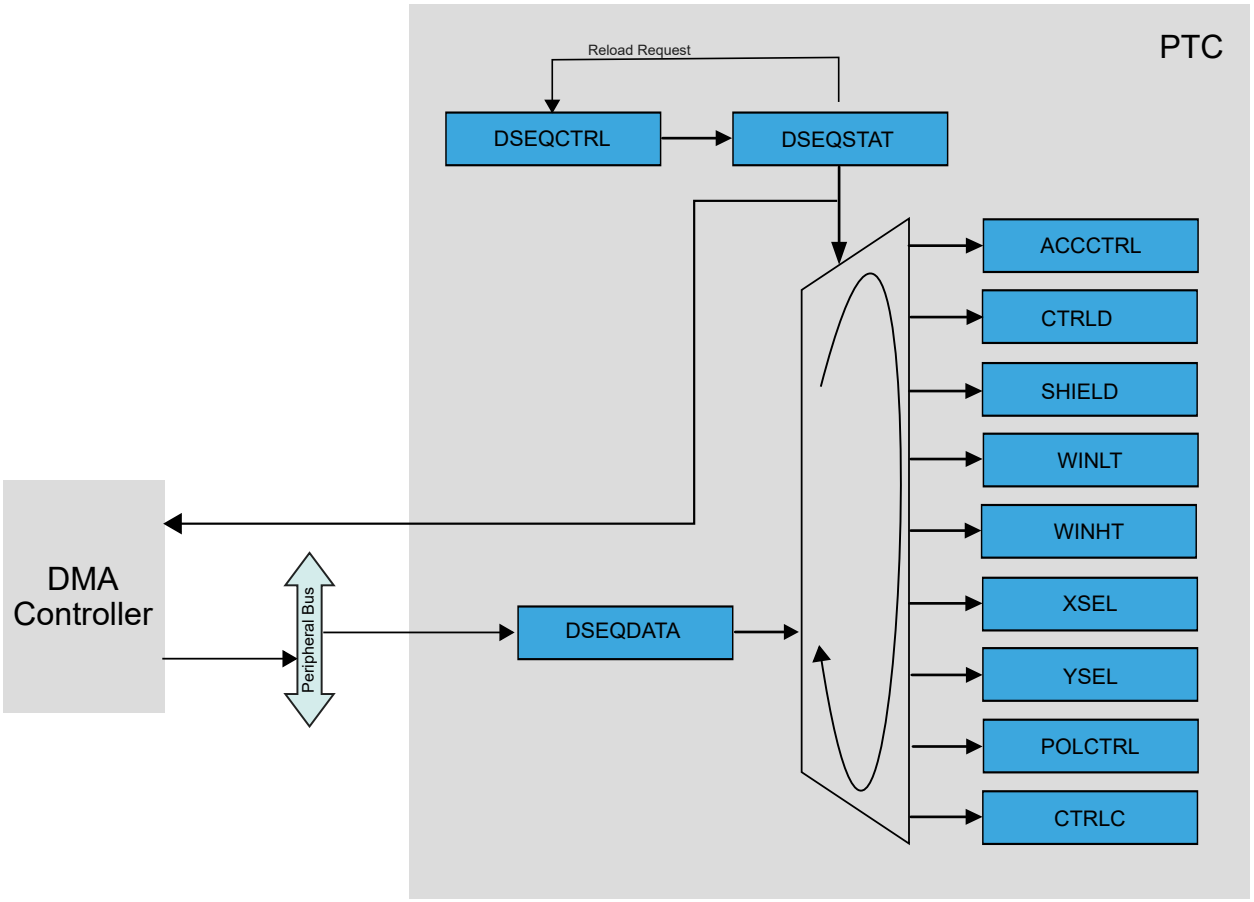
When one of these registers is written, the data is stored in the corresponding buffer as long as the current conversion is not completed, and the corresponding busy status will be set in the


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

Synchronization Busy register (SYNCBUSY). When a new RES is available, data stored in the buffer registers will be transferred to the PTC and a new conversion can start.

#### 39.6.4.5. DMA Sequencing

The PTC can sequence a series of conversion. When enabled, a set of PTC configuration registers can be automatically refreshed using the DMA controller.




**Enable the Sequencing**

The sequencing is enabled when at least one bit in the DMA Sequence Control register (DSEQCTRL) is set to one.

When set, the BUSY status bit in the DMA Sequential Status register (DSEQSTAT.BUSY) is set to one.

**Disable the Sequencing**

The sequencing is disabled when at least one of the following conditions is valid:

- Disable the PTC (CTRLA.ENABLE = 0)
- Reset the PTC (CTRLA.SWRST = 1)
- The DMA Sequence Control register (DSEQCTRL) is written to zero and the ongoing DMA sequence is completed

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.

**Restarting the Sequencing**

When the DSEQSTOP bit is set (CTRLD.DSEQSTOP = 1) and the sequence is disabled (DSEQSTAT.BUSY = 0), it is possible to restart the sequencing by enabling one of the following conditions:


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Clear the DSEQSTOP bit in the Control D register (CTRLD.DSEQSTOP = 0).
- Enable the DSEQR event (EVCTRL.DSEQREI). The sequence will restart when the DMA Sequencing Restart event is received.

### DMA Sequencing Operation

Each PTC register, 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 and the 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 PTC is idle or a new conversion starts, and one of the following conditions are true:

- The Control D bit in the DMA Sequential Control register is set (DSEQCTRL.CTRLD = 1)
- The Window Monitor High Threshold or Window Monitor Low Threshold bits in the DMA Sequential Control register is set (DSEQCTRL.WINHT = 1 orDSEQCTRL.WINLT = 1)
- The X-line Selection bits in the DMA Sequential Control register is set (DSEQCTRL.XSEL0 = 1or DSEQCTRL.XSEL1 = 1)
- The Y-line Selection bits in the DMA Sequential Control register is set (DSEQCTRL.YSEL0 = 1 or DSEQCTRL.YSEL1 = 1)
- The Polarity Control bits in the DMA Sequential Control register is set (DSEQCTRL.POL0 = 1or DSEQCTRL.POL1 = 1)
- The Compensation Capacitor, Integration Capacitor, or Charge Share Delay bits in the DMA Sequential Control register is set (DSEQCTRL.CC = 1, DSEQCTRL.CI = 1 or DSEQCTRL.CSD = 1)

When received, the DMA data must be written to the DSEQDATA register only. Only word (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 PTC registers refresh when the DSEQDATA register is written by the DMA.

**Table 39-5.** DSEQ Trigger Generation and Internal PTC Register Updates

| Condition | Value | Action when DMA writes to DSEQDATA |
|---|---|---|
| DSEQSTAT.CTRLD | 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-bits) must be reserved in the memory<br>• CTRLD.DSEQDATA<23:16> if DSEQSTAT.CTRLD = 1 |
| DSEQSTAT.WINLT \| DSEQSTAT.WINHT | 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-bits) must be reserved in the memory<br>• WINLT.DSEQDATA<15:0> if DSEQSTAT.WINLT = 1<br>• WINHT.DSEQDATA<31:16> if DSEQSTAT.WINHT = 1 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 39-5.** DSEQ Trigger Generation and Internal PTC Register Updates (continued)

| Condition | Value | Action when DMA writes to DSEQDATA |
|---|---|---|
| DSEQSTAT.XSEL0 \| DSEQSTAT.XSEL1 | 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-bits) must be reserved in the memory<br>• XSEL[15:0] <- DSEQDATA[15:0] if DSEQSTAT.XSEL0 = 1<br>• XSEL[31:16] <- DSEQDATA[31:16] if DSEQSTAT.XSEL1 = 1 |
| DSEQSTAT.YSEL0 \| DSEQSTAT.YSEL1 | 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-bits) must be reserved in the memory<br>• YSEL[15:0] <- DSEQDATA[15:0] if DSEQSTAT.YSEL0 = 1<br>• YSEL[31:16] <- DSEQDATA[31:16] if DSEQSTAT.YSEL1 = 1 |
| DSEQSTAT.POL0 \| DSEQSTAT.POL1 | 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-bits) must be reserved in the memory<br>• POLCTRL[15:0] <- DSEQDATA[15:0] if DSEQSTAT.POL0 = 1<br>• POLCTRL[31:16] <- DSEQDATA[31:16] if DSEQSTAT.POL1 = 1 |
| DSEQSTAT.CC \| DSEQSTAT.CI \| DSEQSTAT.CSD | 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-bits) must be reserved in the memory<br>• CTRLC[15:0] <- DSEQDATA[15:0] if DSEQSTAT.CC = 1<br>• CTRLC[23:16] <- DSEQDATA[23:16] if DSEQSTAT.CI = 1<br>• CTRLC[31:24] <- DSEQDATA[31:24] if DSEQSTAT.CSD = 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 PTC conversion starts.

### DMA Descriptor Setup and Data Memory Organization

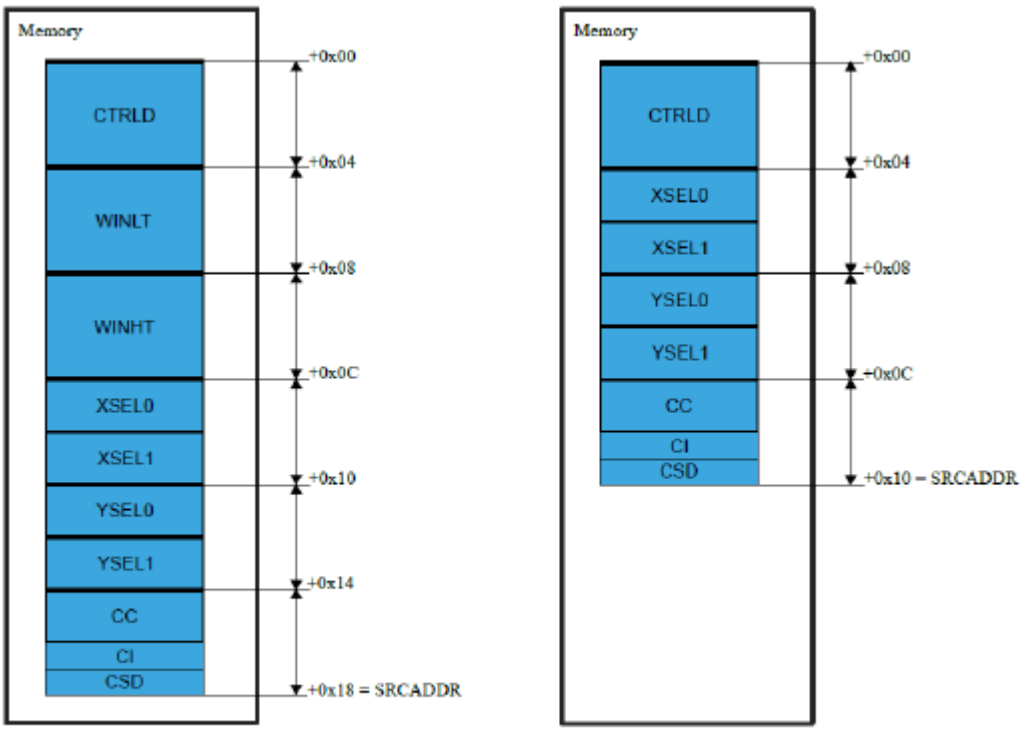
When the DMA sequencing is enabled, the DMA must be configured as given below:

- Select 32-bit beat size transfer (BTCTRL.BEATSIZE = WORD)
- Enable the source address increment options (BTCTRL.SRCINC = 1, BTCTRL.STEPSEL = SRC, BTCTRL.STEPSIZE = X1)
- Disable the destination address increment (BTCTRL.DSTINC = 0)
- Set the block transfer count value (BTCNT)
- Set the block transfer source address (SRCADDR). The address corresponds to the memory section from where the DMA reads the data


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- Select the PTC DSEQDATA address as value for the block transfer destination address (DSTADDR = DSEQDATA address)
- Select the channel single transfer type (CHCTRL.BURSTLEN = SINGLE)
- Select the channel burst trigger action(CHCTRLA.TRIGACT = BURST)
- Select the PTC DMA sequencing trigger as channel trigger source (CHCTRLA.TRIGSRC = DSEQ)
- Enable optional channel interrupts (CHINTENSET)
- Enable the corresponding DMA channel (CHCTRLA.ENABLE)

When a PTC condition is enabled to trigger a DMA transfer, one word (32-bit) will be read by the DMA from the memory source location, because the source address is incrementing by one, the data memory must be organized in a contiguous memory area. As consequence, if PTC group of registers does not generate any DMA trigger, no data must be reserved in the memory area for this register group. The figure below is an example of memory organization when all PTC registers are part of the sequence, and a second example where the WINLT and WINHT registers are not part of the sequence.




All registers are in the sequence

WINLT / WINUT registers are not in the sequence

**Automatic Start Conversion**

### 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 PTC conversion by writing one to the AUTOSTART bit in the DSEQCTRL register (DSEQCTRL.AUTOSTART). When set, the PTC automatically starts a new conversion when a DMA sequence is complete.

**Notes:**

1. If averaging is enabled, the new conversion automatically starts only when the previous RESULT is available (averaging operation is complete).
2. If the conversion is triggered by event (EVCTRL.STCONVEI = 1), the automatic start conversion is disabled and the AUTOSTART settings are ignored.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.6.5. DMA OPERATION

The PTC generates the following DMA request:

- End Of Conversion (EOC) – 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 details, refer to *DMA Sequencing*.

### 39.6.6. Interrupts

The PTC has the following interrupt sources:

- End of Conversion (EOC)
- Window Monitor (WCOMP)
- Analog Core Ready (ACRRDY)

These interrupts are asynchronous wake-up sources. Refer to the *Sleep Mode Controller* for additional information.

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, interrupt is disabled, or PTC is reset. Refer to the INTFLAG register for details on how to clear interrupt flags.

All interrupt requests from the peripheral are ORed together in this macro level to generate one or more combined interrupt requests to the Device Nested Vector Interrupt Controller (NVIC). The user must read the INTFLAG register to determine which interrupt condition is present.

### 39.6.7. Events

The PTC can generate the following output events:

- End Of Conversion (EOC): Generated when the conversion is complete and the result is available. Refer to EVCTRL for details.
- Window Monitor (WCOMP): Generated when the window monitor condition match.

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 PTC can take the following actions on an input event:

- Start a conversion
- Restart DMA sequencing

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 PTC uses only asynchronous events, so the asynchronous Event System channel path must be configured. By default, the PTC will detect a rising edge on the incoming event. If the PTC 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).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 39.6.8. Sleep Mode Operation

The ONDEMAND and RUNSTDBY bits in the Control A register (CTRLA) control the behavior of the PTC during Standby Sleep mode, in cases where the PTC is enabled (CTRLA.ENABLE = 1).

**Note:** When CTRLA.ONDEMAND = 1, the analog block is power 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 39-6.** PTC Sleep Behavior

| CTRLA.RUNSTDBY | CTRLA.ONDEMAND | CTRLA.ENABLE | Description |
|---|---|---|---|
| x | x | 0 | Disabled |
| 0 | 0 | 1 | Run in all sleep modes except Standby mode. |
| 0 | 1 | 1 | Run in all sleep modes on request, except Standby mode. |
| 1 | 0 | 1 | Run in all sleep modes. |
| 1 | 1 | 1 | Run in all sleep modes on request. |

### 39.6.9. Synchronization

Due to the asynchronous nature between the main clock domain and the peripheral clock domains, some registers must be synchronized when written or read.

The following bits are synchronized when written:

- 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 Control B register (CTRLB)
- The Control C register (CTRLC)
- The Control D register (CTRLD)
- The Software Trigger register (SWTRIG)
- The Window Monitor Low Threshold register (WINLT)
- The Window Monitor High Threshold register (WINHT)
- The X-line selection register (XSEL)
- The Y-line selection register (YSEL)
- The Polarity Control register (POLCTRL)

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.

## 39.7. Peripheral Dependencies

**Table 39-7.** Peripheral Dependencies

| Peripheral Name | Base Address | NVIC IRQ Index:Source | MCLK AHB/APB Clock Enable Bus:Register:Bit | GCLK Peripheral Channel Control Register | PAC Peripheral Identifier PERID:Register:Bit | DMA Trigger Index:Source | Event System Type:Event: Register:Path | Power Domain |
|---|---|---|---|---|---|---|---|---|
| PTC | 0x4482C000 | 133 : EOC/ WCOMP/ ACRRDY | **APB:** CLKMSK[1] MASK19 | PCHCTRL[35] | 41 STATUS1 PERID9 | 38 : EOC<br>39 : SEQ<br>40 : WCOMP | Generators:<br>**EOC:** CHANNEL[81]<br>**WCOMP:** CHANNEL[82]<br>Users:<br>**DSEQR:** USER[51],Async<br>**STCONV:** USER[52],Async | PD_CORE_SW |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 40. Electrical Characteristics

## 40.1. Absolute Maximum Electrical Characteristics

Absolute maximum ratings are listed below. Exposure to these maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at these or any other conditions, above the parameters indicated in the operation listings of this specification, is not implied.

**Table 40-1.** Absolute Maximum Electrical Characteristics

| | |
|---|---|
| Ambient temperature under bias | -40°C to +85°C |
| Storage temperature | -65°C to 150°C |
| Voltage on VDD with respect to VSS | -0.3V to 4.03V |
| Voltage on any pin that is not 5V tolerant, with respect to VSS (3) | -0.3V to (VDDIO+0.3V) |
| Voltage on any 5V tolerant pin with respect to VSS when VDDIO ≥ 1.9V (3) | -0.3V to 5.5V |
| Voltage on any 5V tolerant pin with respect to VSS when VDDIO < 1.9V (3) | -0.3V to (VDDIO+0.3V) |
| Voltage on USBFS D+ or D- pin with respect to VSS | -0.3V to (VDDIO+0.3V) |
| Voltage on VREFH with respect to VSS | -0.3V to +(AVDD+0.3V) |
| Maximum current out of a VSS pins (2) | 200 mA |
| Maximum current into a VDDIO pins (2) | 200 mA |
| Maximum output current sourced/sunk by any 4x I/O pin | 15 mA |
| Maximum output current sourced/sunk by any 8x I/O pin | 25 mA |
| Maximum output current sunk/sourced by any port cluster (4) | 200 mA |
| Maximum Junction Temperature | +105°C |
| **ESD qualification:** | |
| Human Body Model (HBM) per JESD22-A114 | 4000V |
| Charged Device Model (CDM) (ANSI/ESD STM 5.3.1) | 1000V |

**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 the device reliability.
2. Maximum allowable current is a function of device maximum power dissipation.
3. See the Table 40-17 for the 5V tolerant pins and 4x/8x pins.
4. When applying higher or lower voltage than those specified on the IO pins, refer to DI_19/DI_21 to respect the maximum injection current specification.
5. Refer to the Table 46-17 for the I/O pin Drive Strength.
6. This maximum value is specific to continuous DC current. For I/O pin AC current characteristics, refer to the *I/O Pin Electrical Specifications* section.
7. A cluster is a group of I/O pins bounded by power/ground pins. Refer to the following table *I/O Clusters*.

**Table 40-2.** I/O Clusters

| I/O Cluster | Pins |
|---|---|
| 1 | PD06, PD07, PD19, PD04, PD05, PD08, PD20, PD09, PA00, PA14, PA01, PA15, PA02 |
| 2 | PA10, PA11, PA12, PA16, PA17, PA18, PA03, PA04, PA05, PA06, PA13 |
| 3 | PA07, PA08, PA09, PA20, PA19, PB00, PB07, PB01, PB02 |
| 4 | PB11, PB12, PB13, PB14, PB03, PB04, PB05, PB08, PB09, PB06 |
| 5 | PB15, PB16, PB17, PB10, PC00, PC11, PC12, PC01, PC13 |
| 6 | PC02, PC14, PC15, PC03, PC16, PC09, PC04, PC17, PC10 |
| 7 | PC18, PC19, PC05, PC06, PC07, PC08, PD00, PC10, PC14, PD11, PD01 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-2.** I/O Clusters (continued)

| I/O Cluster | Pins |
|---|---|
| 8 | PD02, PD03, PD15, PD16, PD17, PD12, PD18, PD13 |

## 40.2. Operating Frequencies and Thermal Specifications

**Table 40-3.** Operating Frequency vs Voltage

| Param. No. | VDDIOx, AVDD Range | Temp. Range (in °C) | Max MCU Frequency | Comments |
|---|---|---|---|---|
| DC_5 | 1.71 to 3.63V | -40°C to +85°C | FCLK_1 MHz | Industrial |

**Table 40-4.** MCU Thermal Operating Conditions

| Rating | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Industrial Temperature Devices Operating Ambient Temperature Range Operating Junction Temperature Range | TA<br>TJ | -40<br>— | — | 85<br>105 | °C<br>°C |
| Power Dissipation:<br>Internal Chip Power Dissipation:<br>PINT = (VDDIOx x (IDD –∑ IOHVDDIOx))+ (AVDD x (IDDANA - ∑ IOHAVDD)) + (VDDREG x IDDREG)<br>I/O Pin Power Dissipation:<br>PI/O = ∑ ((VDDIOx – VOHVDDIOx) x IOHVDDIOx) + ∑ (VOL x IOLVDDIOx) + ∑ ((AVDD – VOHAVDD) x IOHAVDD) + ∑ (VOL x IOLAVDD) | PD | PINT + PI/O | | | W |
| Maximum Allowed Power Dissipation | PDMAX | (TJ – TA)/θJA | | | W |

**Table 40-5.** Thermal Packaging Characteristics (1)

| Characteristics | Symbol | Typ. | Max. | Unit |
|---|---|---|---|---|
| Thermal Resistance, 48-pin TQFP (7x7x1 mm) Package | θJA | 47.0 | — | °C/W |
| Thermal Resistance, 64-pin TQFP (10x10x1 mm) Package | θJA | 44.8 | — | °C/W |
| Thermal Resistance, 100-pin TQFP (14x14x1 mm) Package | θJA | 41.1 | — | °C/W |
| Thermal Resistance, 48-pin VQFN (7x7x0.9 mm) Package | θJA | 22.3 | — | °C/W |
| Thermal Resistance, 64-pin VQFN (9x9x1 mm) Package | θJA | 20.5 | — | °C/W |

**Note:**

1. Junction to ambient thermal resistance, Theta-JA (θJA) numbers are achieved by package simulations.

## 40.3. Power Supply

**Table 40-6.** Power Supply DC Electrical Specifications

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| REG_5 | VDDIO_CIN (4) | VDDIO Input Bypass parallel Capacitor pair | 33 | — | — | µF | Bulk Ceramic or solid Tantalum with ESR <0.5Ω (5) |
| | | | 100 | — | — | nF | Ceramic X7R with ESR <0.5Ω on all VDDIOx pins (5) |
| REG_6 | VDDREG_CIN (4) | VDDREG Input Bypass parallel Capacitor pair | 33 | — | — | µF | Bulk Ceramic or solid Tantalum with ESR <0.5Ω (5) |
| | | | 100 | — | — | nF | Ceramic X7R with ESR <0.5Ω on all VDDIOx pins (5) |
| REG_15 | USERLDO_COUT | User LDO output capacitor | 10 | — | — | uF | Ceramic X7R with ESR <0.5Ω |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-6.** Power Supply DC Electrical Specifications (continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Conditions** |
| REG_17 | AVDD_CIN (4) | AVDD Input Bypass parallel Capacitor pair | 10 | — | — | µF | Bulk Ceramic or solid Tantalum with ESR <0.5Ω |
| | | | 100 | — | — | nF | Ceramic X7R with ESR <0.5Ω |
| REG_23 | AVDD_LEXT (1) | AVDD series Ferrite Bead DCR (DC Resistance) | — | — | 0.1 | Ω | ≥600 Ω @ 100 MHz |
| REG_25 | | Ferrite Bead current Rating | 500 | — | — | mA | — |
| REG_37 | VDDIO (2) | VDDIO Input Voltage Range | 1.71 | 3.3 | 3.63 | V | — |
| REG_39 | AVDD (2) | AVDD Input Voltage Range | 1.71 | 3.3 | 3.63 | V | — |
| REG_40 | VDDREG (3) | VDDREG Input Voltage Range | 1.71 | 3.3 | 3.63 | V | — |
| REG_43 | SVDD_R | VDDIO/AVDD/VDDREG Rise Ramp Rate to Ensure Internal Power-on Reset Signal | 0.00000033 | — | 0.33 | V/µs | Failure to meet this specification may lead to start-up or unexpected behaviors |
| REG_45 | VPOR | Power-on Reset | 1.44 | 1.5 | 1.59 | V | VDD Power-down |
| REG_47 | VDDIO / AVDD BOR (5) | VDDIO / AVDD Brown-out Reset Thresholds | 1.63 | — | 1.7 | V | BOR_TRIP_VDDx = 0x0<br>HYST_BOR_VDDx = 0x0 |
| | | | 2.11 | — | 2.24 | V | BOR_TRIP_VDDx = 0x1<br>HYST_BOR_VDDx = 0x0 |
| | | | 2.49 | — | 2.68 | V | BOR_TRIP_VDDx = 0x2<br>HYST_BOR_VDDx = 0x0 |
| | | | 2.75 | — | 2.97 | V | BOR_TRIP_VDDx = 0x3<br>HYST_BOR_VDDx = 0x0 |
| | | | 1.61 | — | 1.7 | V | BOR_TRIP_VDDx = 0x0<br>HYST_BOR_VDDx = 0x1 |
| | | | 2.05 | — | 2.24 | V | BOR_TRIP_VDDx = 0x1<br>HYST_BOR_VDDx = 0x1 |
| | | | 2.41 | — | 2.68 | V | BOR_TRIP_VDDx = 0x2<br>HYST_BOR_VDDx = 0x1 |
| | | | 2.62 | — | 2.97 | V | BOR_TRIP_VDDx = 0x3<br>HYST_BOR_VDDx = 0x1 |
| REG_49 | VDDREG BOR (5) | VDDREG Brown-out Reset Thresholds | 1.63 | — | 1.7 | V | HYST_BOR_VDDREG = 0x0 (6) |
| | | | 1.61 | — | 1.7 | V | HYST_BOR_VDDREG = 0x1 (6) |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-6.** Power Supply DC Electrical Specifications (continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| REG_51 | VDDIO / AVDD / VDDREG DCBOR (7) | VDDIO / AVDD / VDDREG Duty Cycled BOR Thresholds | 1.43 | — | 1.75 | V | BOR_TRIP = 0x0<br>BOR_HYS = 0x0 |
| | | | 1.87 | — | 2.28 | V | BOR_TRIP = 0x1<br>BOR_HYS = 0x0 |
| | | | 2.22 | — | 2.75 | V | BOR_TRIP = 0x2<br>BOR_HYS = 0x0 |
| | | | 2.43 | — | 3.01 | V | BOR_TRIP = 0x3<br>BOR_HYS = 0x0 |
| | | | 1.4 | — | 1.75 | V | BOR_TRIP = 0x0<br>BOR_HYS = 0x1 |
| | | | 1.81 | — | 2.28 | V | BOR_TRIP = 0x1<br>BOR_HYS = 0x1 |
| | | | 2.13 | — | 2.75 | V | BOR_TRIP = 0x2<br>BOR_HYS = 0x1 |
| | | | 2.27 | — | 3.01 | V | BOR_TRIP = 0x3<br>BOR_HYS = 0x1 |
| REG_52 | LVD (8) | VDDIO Low Voltage Detector Thresholds | 1.667 | — | 1.76 | V | LVD.LEVEL = 0x0 |
| | | | 1.741 | — | 1.839 | V | LVD.LEVEL = 0x1 |
| | | | 1.821 | — | 1.924 | V | LVD.LEVEL = 0x2 |
| | | | 1.959 | — | 2.071 | V | LVD.LEVEL = 0x3 |
| | | | 2.062 | — | 2.181 | V | LVD.LEVEL = 0x4 |
| | | | 2.118 | — | 2.241 | V | LVD.LEVEL = 0x5 |
| | | | 2.177 | — | 2.304 | V | LVD.LEVEL = 0x6 |
| | | | 2.255 | — | 2.37 | V | LVD.LEVEL = 0x7 |
| | | | 2.321 | — | 2.44 | V | LVD.LEVEL = 0x8 |
| | | | 2.43 | — | 2.554 | V | LVD.LEVEL = 0x9 |
| | | | 2.635 | — | 2.771 | V | LVD.LEVEL = 0xA |
| | | | 2.726 | — | 2.869 | V | LVD.LEVEL = 0xB |
| | | | 2.827 | — | 2.974 | V | LVD.LEVEL = 0xC |
| | | | 2.933 | — | 3.084 | V | LVD.LEVEL = 0xD |
| | | | 3.248 | — | 3.405 | V | LVD.LEVEL = 0xE |
| | | | 3.543 | — | 3.711 | V | LVD.LEVEL = 0xF |
| REG_53 | TRST | External RESET valid active pulse width | 2 | — | — | µs | Minimum reset active time to guarantee MCU reset |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-6.** Power Supply DC Electrical Specifications (continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |

**Notes:**

1. Ferrite Bead ISAT(min) ≥ (IDDANA(max) * 1.15).
2. VDDIO and AVDD must be at the same voltage level.
3. VDDREG voltage must be equal or lower than VDDIO.
4. All bypass caps should be located immediately adjacent to pins and on the same side of the PCB as the MCU. Each primary power supply group VDDIO, AVDD , VDDREG should have one bulk capacitor and all power pins everywhere a 100 nF bypass cap.
5. Voltages below the minimum BOR threshold will result in a device reset, except for the BOR_VDDUSB that can cause an interrupt. Voltages above the maximum BOR threshold will allow the device starting-up.
6. Overall functional device operation at VBORMIN < VDD < VDDMIN is guaranteed, but not characterized. Device will function with degraded performances below VDDMIN.
7. Voltages below the minimum DCBOR threshold will result in a device reset.
8. Voltages below the minimum LVD threshold will result in a falling detection. Voltages above the maximum LVD threshold will result in a rising detection. Enabling the LVD when VDDIO is between the minimum or maximum threshold may result in unexpected behavior.

## 40.4. Active Power

**Table 40-7.** MCU Active Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Clock/Freq | Power Pin | Typ. (1) | Max. | Units | Conditions |
| APWR_1 | IDD_ACTIVE (2,3) | MCU IDD in Active mode | PLL_72_MHz | VDDIO = AVDD = 3.3V | 2.6 | 3 | mA | PLL runs at 72 MHz, uses XOSC with 48 MHz crystal as reference clock |
| APWR_3 | | | | VDDREG = 3.3V | 11.1 | 17.5 | mA | |
| APWR_5 | | | DFLL _48_MHz | VDDIO = AVDD = 3.3V | 0.8 | 2 | mA | DFLL in open loop |
| APWR_7 | | | | VDDREG = 3.3V | 4.9 | 12 | mA | |
| APWR_9 | | | XOSC _48_MHz | VDDIO = AVDD = 3.3V | 1.7 | 2.5 | mA | 48 MHz crystal |
| APWR_11 | | | | VDDREG = 3.3V | 5.4 | 11 | mA | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-7.** MCU Active Current Consumption DC Electrical Specifications (continued)

| DC CHARACTERISTICS | | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Clock/Freq | Power Pin | Typ. (1) | Max. | Units | Conditions |

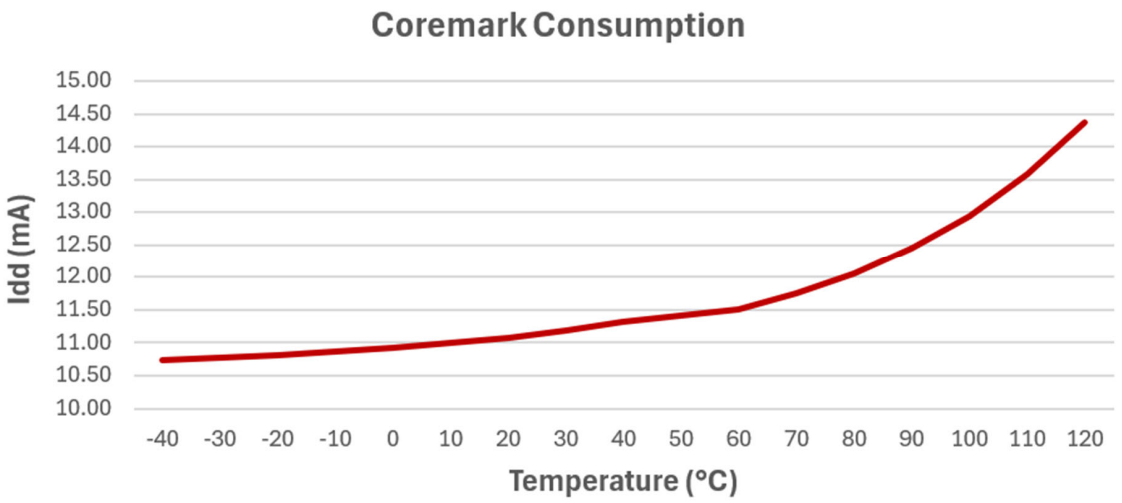
**Notes:**

1. Typical values at 25°C only.
2. Conditions:
   - RTC running on ULP32K for CoreMark® ticks count
   - No other peripheral modules are operating (i.e., all peripherals except RTC inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled
3. MCU Running CoreMark® Test Suite.

Operating CONDITIONS

- VDDIO = 3.3V
- PLL_48_Mhz

**Figure 40-1.** Power Consumption over Temperature in Active Mode (Typical values for guidance only, not tested)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.5. Idle Power

**Table 40-8.** MCU Idle Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Clock/Freq | Power Pin | Typ. (1) | Max. | Units | Conditions |
| IPWR_1 | IDD_IDLE (2) | MCU IDD in Idle mode | PLL_72_MHz | VDDIO = AVDD = 3.3V | 1.4 | 1.4 | mA | PLL runs at 72 MHz, uses XOSC with 48 MHz crystal as reference clock |
| IPWR_3 | | | | VDDREG = 3.3V | 5.6 | 11.6 | mA | |
| IPWR_5 | | | DFLL _48_MHz | VDDIO = AVDD = 3.3V | 0.8 | 0.9 | mA | DFLL in open loop |
| IPWR_7 | | | | VDDREG = 3.3V | 1.8 | 10 | mA | |
| IPWR_9 | | | XOSC _48_MHz | VDDIO = AVDD = 3.3V | 1.4 | 1.4 | mA | 48 MHz crystal |
| IPWR_11 | | | | VDDREG = 3.3V | 2.1 | 8 | mA | |

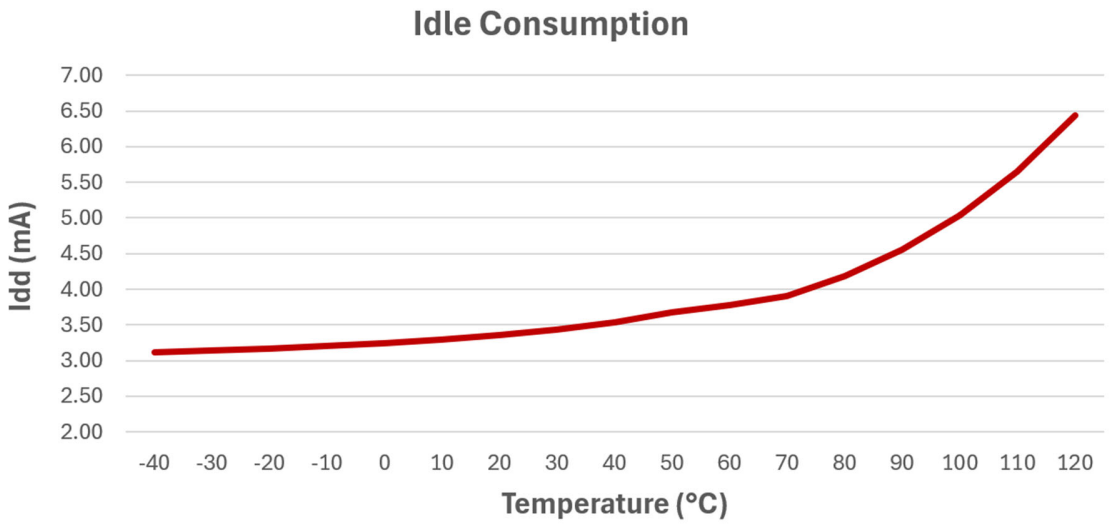
**Notes:**

1. Typical values at 25°C only
2. Conditions:
   - No peripheral modules are operating (i.e., all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

Operating Conditions:

- VDDIO = 3.3V
- PLL_48_MHz

**Figure 40-2.** Power Consumption over Temperature in Idle mode (Typical values for guidance only, not characterized over process or voltage).





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.6. Standby Power

**Table 40-9.** MCU Standby Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Power Pin | Typ. (1) | Max. | Units | Conditions |
| SPWR_5 | IDD_STANDBY (2) | MCU IDD in Standby mode w/VDDCORE_x = 0.8V | VDDIO = AVDD = 3.3V | 0.22 | 0.28 | mA | RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1),<br>NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0),<br>Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1)<br>Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0) |
| SPWR_7 | | | VDDREG = 3.3V | 0.47 | 3 | mA | |
| SPWR_9 | | | VDDIO = VDDANA = 3.3V | 0.13 | 0.18 | mA | RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1),<br>NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0),<br>Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1)<br>Voltage Reference Low Voltage enabled (SUPC.VREF.LPSTDBY = 0x1) |
| SPWR_11 | | | VDDREG = 3.3V | 0.16 | 2.7 | mA | |
| SPWR_17 | | MCU IDD in Standby mode w/VDDCORE_x = 1.2V | VDDIO = AVDD = 3.3V | 0.25 | 0.31 | mA | RAM Low-Power mode enabled (PM.STDBYCFG.LPRAM = 0x1),<br>NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0),<br>Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0)<br>Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0) |
| SPWR_19 | | | VDDREG = 3.3V | 0.7 | 7.7 | mA | |
| SPWR_21 | | | VDDIO = AVDD = 3.3V | 0.16 | 0.21 | mA | RAM Low- Power mode enabled (PM.STDBYCFG.LPRAM = 0x1),<br>NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0),<br>Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0)<br>Voltage Reference Low Voltage enabled (SUPC.VREF.LPSTDBY = 0x1) |
| SPWR_23 | | | VDDREG = 3.3V | 0.33 | 7.5 | mA | |

**Notes:**

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating, unless specified (i.e., all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

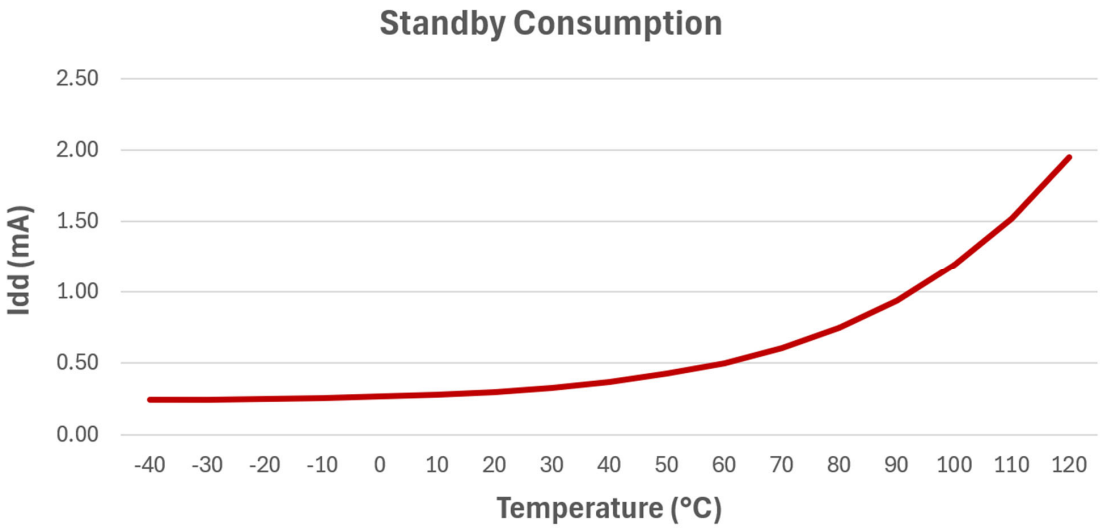
Operating CONDITIONS

- VDDIO = 3.3V
- RAM Low Power mode enabled


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- NVM Power Reduction Mode is Enter Auto Standby
- Voltage Regulators Low Voltage enabled
- Voltage Reference Low Voltage enabled
- No RTC

**Figure 40-3.** Power Consumption over Temperature in Standby Mode (Typical values for guidance only, not tested)




## 40.7. Hibernate Power

**Table 40-10.** MCU Hibernate Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Power Pin** | **Typ. (1)** | **Max.** | **Units** | **Conditions** |
| HPWR_8 | IDD_HIBERNATE (2) | MCU IDD in Hibernate mode w/ VDDCORE_x = 0.8V | VDDIO = VDDANA = 3.3V | 101 | 125 | µA | RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)<br>Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0)<br>RTC running on XOSC32K |
| HPWR_9 | | | VDDREG = 3.3V | 200 | 992 | µA | |
| HPWR_12 | | | VDDIO = AVDD = 3.3V | 19 | 33 | µA | RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)<br>Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1)<br>RTC running on XOSC32K |
| HPWR_13 | | | VDDREG = 3.3V | 48 | 874 | µA | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-10.** MCU Hibernate Current Consumption DC Electrical Specifications (continued)

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Power Pin** | **Typ. (1)** | **Max.** | **Units** | **Conditions** |
| HPWR_18 | IDD_HIBERNATE (2) | MCU IDD in Hibernate mode w/ VDDCORE_x = 1.2V | VDDIO = AVDD = 3.3V | 101 | 125 | µA | RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0) Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0) RTC running on XOSC32K |
| HPWR_19 | | | VDDREG = 3.3V | 292 | 2470 | µA | |
| HPWR_22 | | | VDDIO = AVDD = 3.3V | 19 | 33 | µA | RAM Low- Power mode enabled (PM.HIBCFG.LPRAM = 0x1), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0) Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1) RTC running on XOSC32K |
| HPWR_23 | | | VDDREG = 3.3V | 101 | 2470 | µA | |

**Notes:**

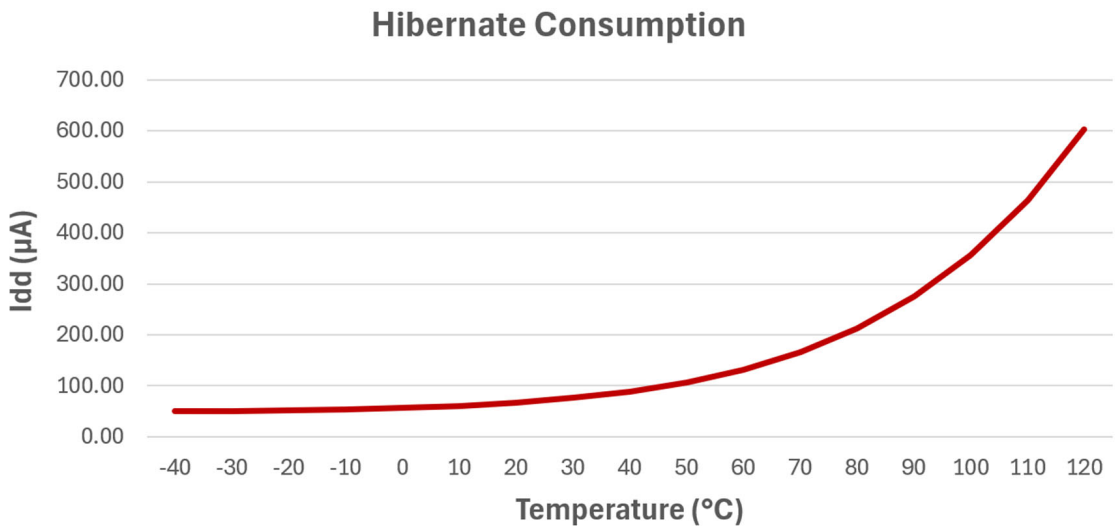
1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating, unless specified (i.e., all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

Operating CONDITIONS

- VDDIO = 3.3V
- Voltage Regulators Low Voltage enabled
- Voltage Reference Low Voltage enabled
- RAM Low Power mode enabled
- No RTC


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-4.** Power Consumption over Temperature in Hibernate Mode (Typical values for guidance only, not tested)




## 40.8. Backup Power

**Table 40-11.** MCU Backup Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Power Pin | Typ (1) | Max. | Units | Conditions |
| BPWR_1 | IDD_BACKUP (2) | MCU IDD in Backup mode powered from VDDIOx | VDDIO = AVDD = 3.3V | 6.3 | 64.6 | µA | RTC running on XOSC32K (32.768 KHz), TRAM enabled |
| BPWR_3 | | | VDDREG = 3.3V | 0.6 | 1 | µA | |

**Notes:**

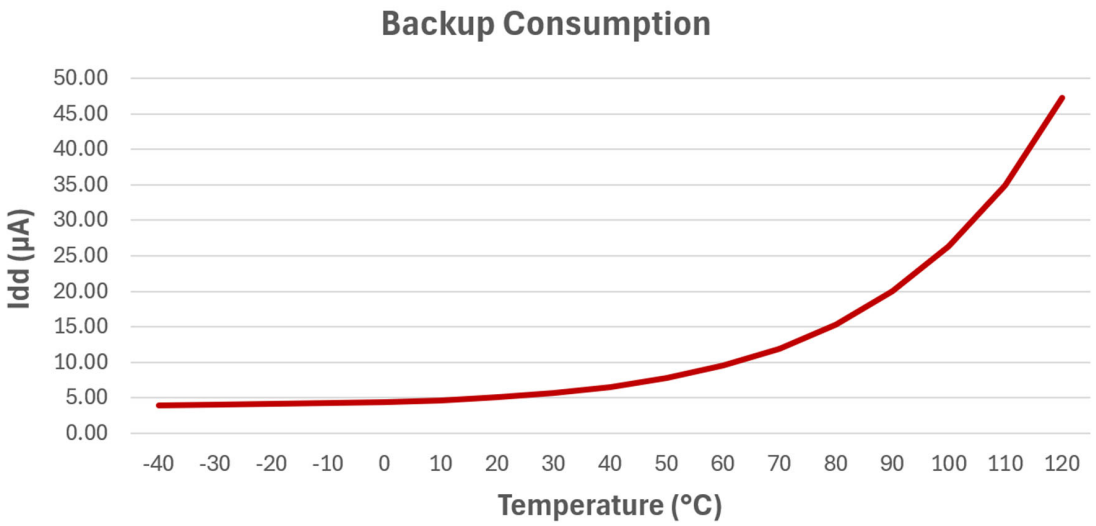
1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating (i.e., all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - CMCC Cache disabled
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

Operating CONDITIONS

- VDDIO = 3.3V
- No RTC


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-5.** Power Consumption over Temperature in Backup Mode (Typical values for guidance only, not tested)




## 40.9. OFF Power

**Table 40-12.** MCU Off Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Power Pin** | **Typ. (1)** | **Max.** | **Units** | **Conditions** |
| OPWR_1 | IDD_OFF (2) | MCU IDD in OFF | VDDIO = AVDD = 3.3V | 1 | 4.4 | µA | In Off mode, the device is entirely powered-off. (SLEEPCFG.SLEEPMODE = OFF), and subsequent execution of the WFI instruction.<br>**Note:** This mode is left by pulling the RESET pin low, or when a power Reset is done. |
| OPWR_3 | | | VDDREG = 3.3V | 0.03 | 0.36 | µA | |

**Notes:**

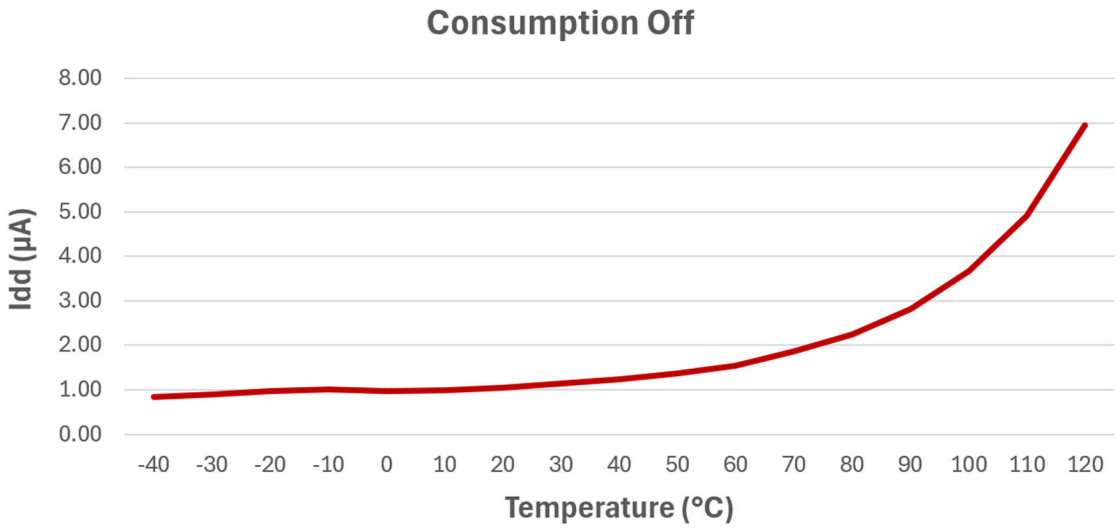
1. Typical values at 25°C only.
2. Conditions:
   - All peripherals inactive unless otherwise specified
   - All clock generation sources disabled unless otherwise specified
   - All I/O pins configured as input pins pulled down or tied to GND

Operating CONDITIONS

- VDDIO = 3.3V


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-6.** Power Consumption over Temperature in Off Mode (Typical values for guidance only, not tested)




## 40.10. Wake-Up Timing

**Table 40-13.** Wake-Up Timings from Low-Power Modes AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| WKUP_1 | WKUP_IDLE | Wake from Idle mode | — | 0.311 | — | µs | |
| WKUP_7 | WKUP_STDBY | Wake from Stnadby mode | — | 12.5 | — | µs | |
| WKUP_11 | WKUP_HIBER | Wake from Hibernate mode | — | 1.66 | — | ms | |
| WKUP_13 | WKUP_BCKUP | Wake from Backup mode | — | 1.63 | — | ms | |
| WKUP_15 | WKUP_OFF | Wake from Off mode | — | 1.40 | — | ms | |

**Note:**
1. VDD = 3.3V, CPU clock = 48 MHz, automatic wait states, Temperature = 25°C.

## 40.11. Peripheral Active Power

**Table 40-14.** Peripheral Active Current DC Electrical Specifications

| DC CHARACTERISTICS (1) | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** |
|---|---|---|---|
| Param. No. | Symbol | Characteristics | Conditions |
| **MODULES/PERIPHERALS ACTIVE CURRENTS ≤1mA** | | | |
| PAI_51 | AC | Analog Comparator Current | 2 Comparators Enabled |
| PAI_52 | BOR | BOR Current | — |
| PAI_53 | CCL | CCL Current | — |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-14.** Peripheral Active Current DC Electrical Specifications (continued)

| DC CHARACTERISTICS [1] | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** |
|---|---|---|---|
| Param. No. | Symbol | Characteristics | Conditions |
| PAI_54 | EIC | EIC Current | — |
| PAI_55 | EVSYS | EVSYS Current | — |
| PAI_56 | FREQM | FREQM Current | — |
| PAI_57 | SERCOM | SERCOM Current | SPI mode, baudrate = 20 MHz, internal loopback |
| PAI_58 | WDT | WDT Current | — |
| **MODULES/PERIPHERALS ACTIVE CURRENTS ≤3mA** | | | |
| PAI_100 | ADC | Analog-to- Digital Converter Current | 12-bit, 4.5 Msps |
| PAI_101 | CAN-FD | CAN-FD Current | 8 Mbps, 10 Mbps |
| PAI_102 | PTC | PTC Current | Mutual Capacitance |
| PAI_103 | TCC | TCC Current | 48 MHz |
| PAI_104 | USB FS | Full-Speed USB Current | — |
| PAI_105 | XOSC | XOSC Current | 48 MHz crystal, AGC Enabled |
| **MODULES/PERIPHERALS ACTIVE CURRENTS ≤10mA** | | | |
| PAI_400 | DMA | DMA Current | — |
| PAI_401 | Flash | Flash Erase/Write | — |
| PAI_402 | PLL | PLL Active Current | PLL output = 72 MHz |

**Note:**

1. These values are for worst case guidance only unless otherwise specified, the actual values depend on user peripheral configuration and operating frequency.

## 40.12. I/O Pin Electrical Characteristics

**Table 40-15.** I/O Pin AC/DC Electrical Specifications

| AC - DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| DI_1 | VIL | Input Low Voltage I/O Pins | GND | — | 0.2*VDDIO | V | VDDIO(min) -to- VDDIO(max) |
| DI_3 | VIH | Input High Voltage Non-5v tolerant I/O Pins | 0.7*VDDIO | — | VDDIO | V | VDDIO(min) -to- VDDIO(max) |
| | | Input High Voltage I/O Pins 5V-tolerant | 0.7*VDDIO | — | 5.5V | V | VDDIO(min) -to- 5.5V |
| DI_5 | VOL | 4x Drive Strength I/O pins (Output Low) | — | — | 0.4 | V | VDDIO < 3.0V @ IOL = 3mA[6] |
| | | | | | | | VDDIO ≥ 3.0V @ IOL = 7mA [6] |
| | | 8x Drive Strength I/O pins (Output Low) | | | | | VDDIO < 3.0V @ IOL = 6mA [6] |
| | | | | | | | VDDIO ≥ 3.0V @ IOL = 10mA [6] |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-15.** I/O Pin AC/DC Electrical Specifications (continued)

| AC - DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Min.** | **Typ** | **Max.** | **Units** | **Conditions** |
| DI_9 | VOH | 4x Drive Strength I/O pins (Output High) | VDDIO-0.6V | — | — | V | VDDIO < 3.0V @ IOH = 5mA(6) |
| | | | | | | | VDDIO ≥ 3.0V @ IOH = 11mA(6) |
| | | 8x Drive Strength I/O pins (Output High) | | | | | VDDIO < 3.0V @ IOH = 9mA(6) |
| | | | | | | | VDDIO ≥ 3.0V @ IOH = 14mA(6) |
| DI_13 | IIL | Input pin leakage current | -1 | — | 1 | µA | GND ≤ VPIN ≤ VDDIO(max) (VPIN = Voltage present on Pin) |
| DI_15 | RPDWN | Internal Pull-Down (DIR = OUT = 0, PULLEN = 1) | 12.6 | — | 14.6 | kΩ | VDDIOx(min) - VDDIO(max) |
| DI_17 | RPUP | Internal Pull-Up (DIR = 0, OUT = PULLEN = 1) | 12 | — | 14.5 | kΩ | |
| DI_19 | IICL | Input Low Injection Current | -5 | — | — | mA | This parameter applies to all I/O pins (1, 4, 5) |
| DI_21 | IICH | Input High Injection Current | — | — | 8 | mA | This parameter applies to all pins, with the exception of 5V tolerant I/O pins. (2, 3, 4, 5). |
| DI_23 | ∑IICT | Total Input Injection Current (sum of all I/O and control pins) Absolute value of \| ∑IICT \| | -69 | — | 69 | mA | Absolute instantaneous sum of all ± input injection currents from all I/O pins. ( \| IICL \| + \| IICH \| ) ≤ ∑IICT |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-15.** I/O Pin AC/DC Electrical Specifications (continued)

| AC - DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| DI_25 | TRISE | I/O pin Rise Time (DRVSTR = 4x) | — | — | 10 | ns | VDDIOx(min), CLOAD = 30 pF(max) |
| DI_27 | TFALL | I/O pin Fall Time (DRVSTR = 4x) | — | — | 9 | ns | |
| DI_29 | TRISE | I/O pin Rise Time (DRVSTR = 8x, SLEW = FAST) | — | — | 5.5 | ns | |
| | | I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW4) | — | — | 21 | ns | |
| | | I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW8) | — | — | 38 | ns | |
| | | I/O pin Rise Time (DRVSTR = 8x, SLEW = SLOW12) | — | — | 54 | ns | |
| DI_31 | TFALL | I/O pin Fall Time (DRVSTR = 8x, SLEW = FAST) | — | — | 5.5 | ns | |
| | | I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW4) | — | — | 24 | ns | |
| | | I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW8) | — | — | 43 | ns | |
| | | I/O pin Fall Time (DRVSTR = 8x, SLEW = SLOW12) | — | — | 61 | ns | |

**Notes:**

1. VIL source < (GND - 0.3). Characterized but not tested.
2. VIH source > (VDDIO + 0.3) for non-5V tolerant pins only.
3. Digital 5V tolerant pins do not have an internal high side Diode to VDDIO, therefore, cannot tolerate any 'positive' input injection current.
4. If the sum of all injection currents are > | ∑IICT | it can affect the ADC results by approximately 4 to 6 counts (i.e., VIH Source > (VDDIO + 0.3) or VIL source < (GND - 0.3)).
5. Any number and the combination of I/O pins not excluded under IICL or IICH conditions are permitted provided the 'absolute instantaneous' sum of the input injection currents from all pins do not exceed the specified ∑IICT limit. To limit the injection current the user must insert a resistor in series RSERIES, (i.e., RS), between input source voltage and device pin. The resistor value is calculated according to:
   - For negative Input voltages less than (GND-0.3): RS ≥ absolute value of | ((VIL source - (GND - 0.3)) / IICL) |
   - For positive input voltages greater than (VDDIO +0.3): RS ≥ ((VIH source - (VDDIO +0.3))/ IICH)
   - For Vpin voltages >VDDIO +0.3 and <GND-0.3 then RS = the larger of the values calculated above
6. IOL/IOH is specific to AC operation. For continuous DC operation characteristics, see the Absolute Maximum Ratings section.
7. Not all I²C operating modes are supported on all pin types. Refer to the following table, I²C Support.
8. See the I/O Drive Strength table for the I/O pin Drive Strength.

**Table 40-16.** I²C Support

| I²C Mode | I/O Pin Drive Strength | |
|---|---|---|
| | 4x | 8x |
| I²C Standard Mode (SM), 100 kHz w/400 pF load | No | Yes |
| I²C Fast Mode (FM), 400 kHz w/400 pF load | No | Yes |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-16.** I²C Support (continued)

| I²C Mode | I/O Pin Drive Strength | |
|---|---|---|
| | 4x | 8x |
| I²C Fast Mode Plus (FM+), 1 MHz w/550 pF load | No | Yes |
| I²C High Speed (HS), 3.4 MHz w/100 pF load | No | Yes |

**Table 40-17.** I/O Drive Strength

| Drive Strength | IO Port/Pad |
|---|---|
| 4x | PA10, PA11, PA12, PA13, PA07, PA08, PA09, PA20, PA19, PB00, PB07, PB01, PB02, PB11, PB12, PB13, PB14, PB08, PB09, PB10, PC09, PC10, PC07, PC08, PD02, PD03, PD12, PD13, PD04, PD05 |
| 8x | PD09, PA00, PA14, PA01, PA15, PA02, PA16, PA17, PA18, PA03, PA04, PA05, PA06, PB03, PB04, PB05, PB06, PB15, PB16, PB17, PC00, PC11, PC12, PC01, PC13, PC02, PC14, PC15, PC03, PC16, PC04, PC17, PC18, PC19, PC05, PC06, PD00, PD10, PD14, PD11, PD01, PD15, PD16, PD17, PD18, PD06, PD07, PD19, PD08, PD20 |

**Figure 40-7.** IOL vs VOL, 4x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

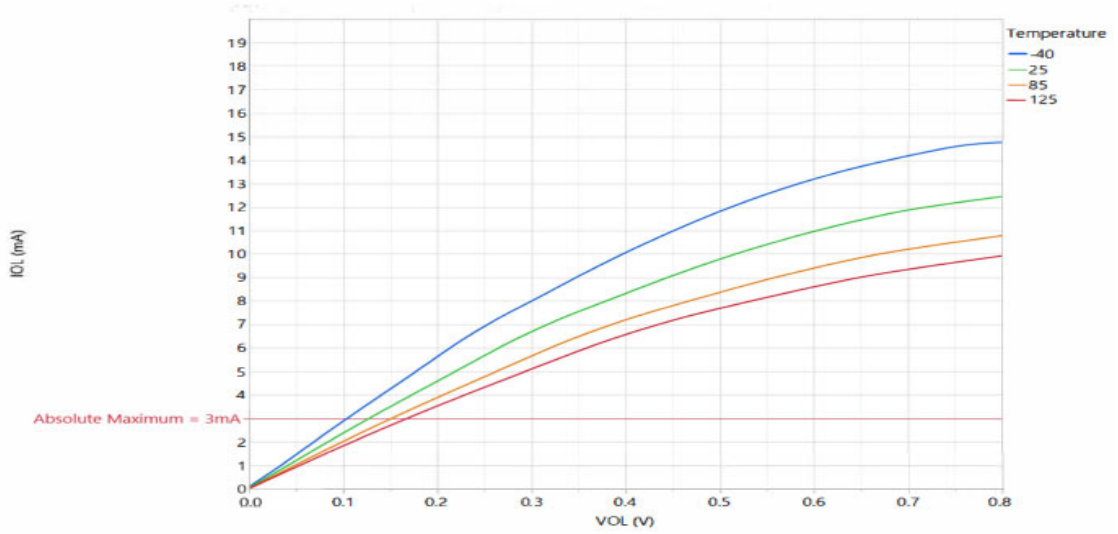
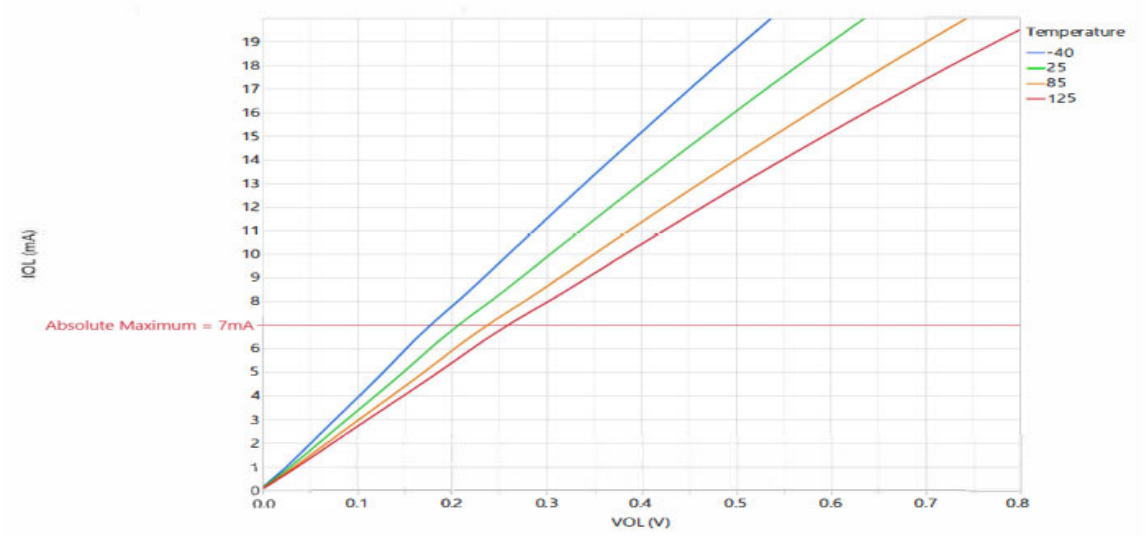



**Figure 40-8.** IOL vs VOL, 4x Drive Strength I/O on Typical parts with VDDIO = 3.0 V





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-9.** IOL vs VOL, 8x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

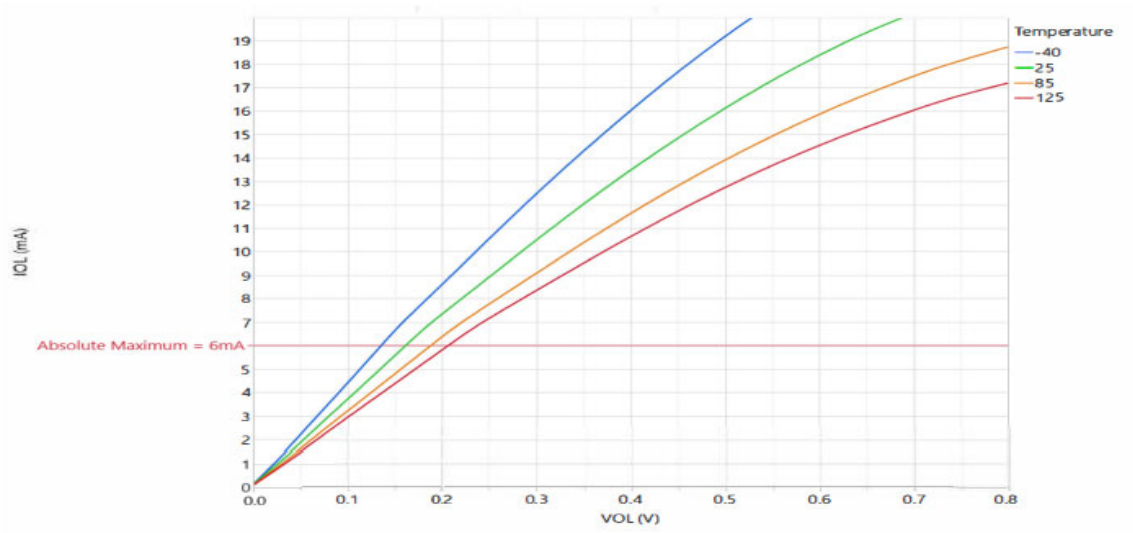
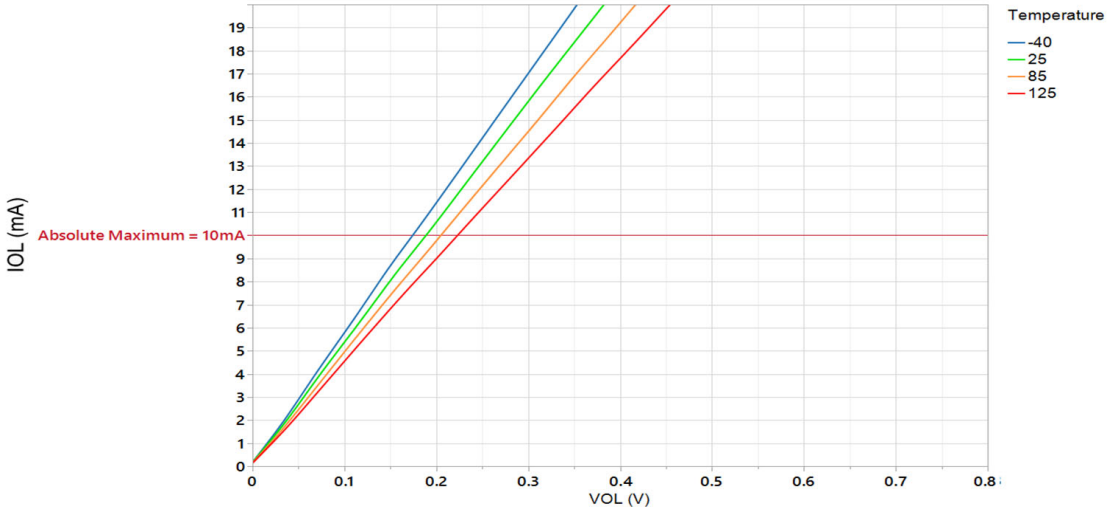



**Figure 40-10.** IOL vs VOL, 8x Drive Strength I/O on Typical parts with VDDIO = 3.0 V





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-11.** IOH vs VOH, 4x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

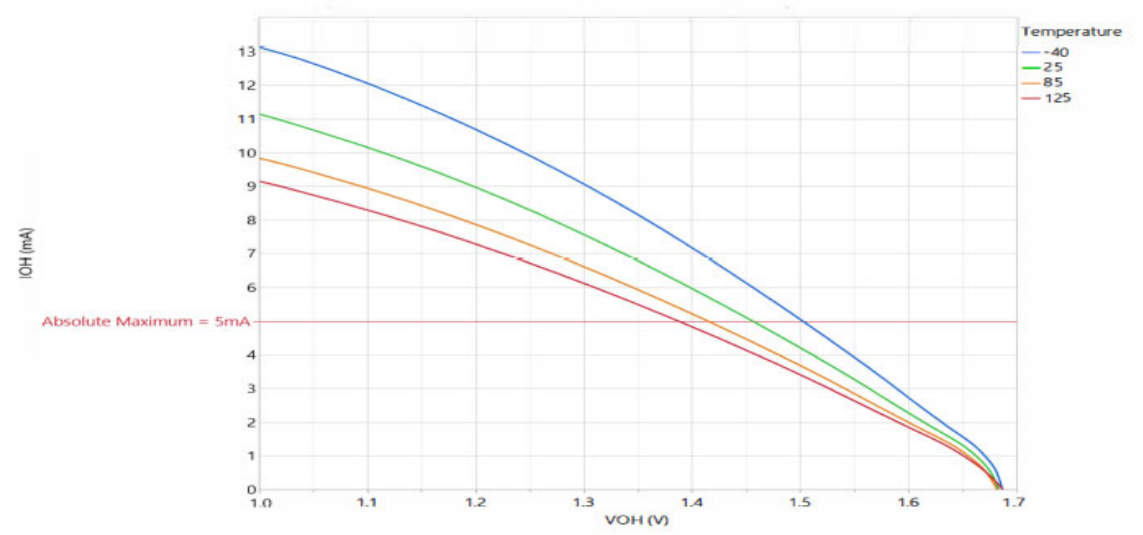
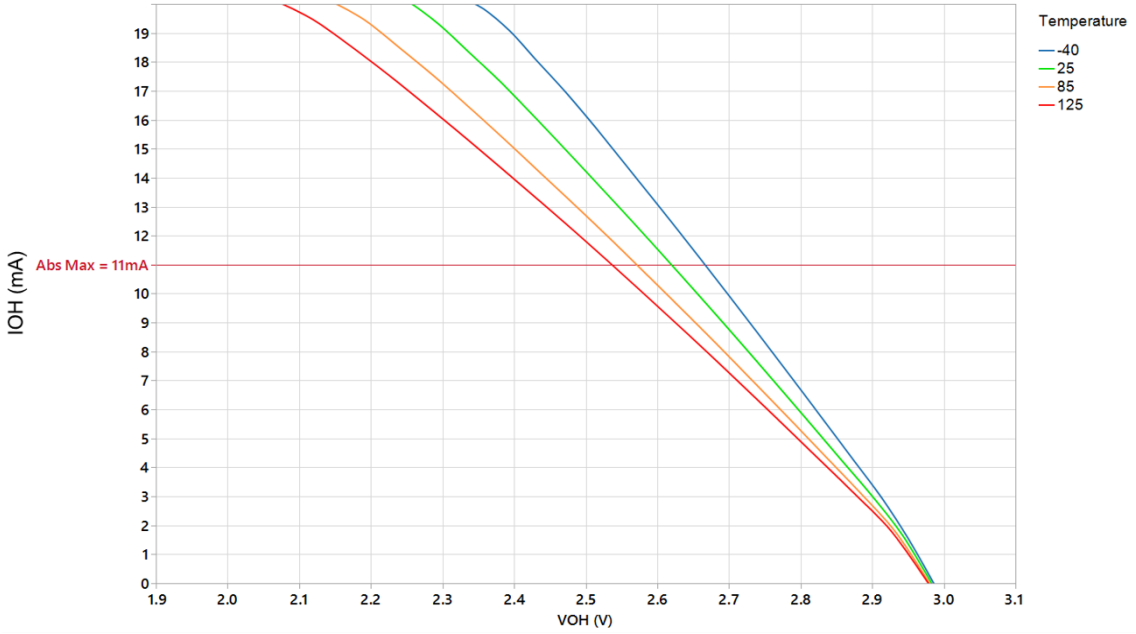



**Figure 40-12.** IOH vs VOH, 4x Drive Strength I/O on Typical parts with VDDIO = 3.0 V





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-13.** IOH vs VOH, 8x Drive Strength I/O on Typical parts with VDDIO = 1.71 V

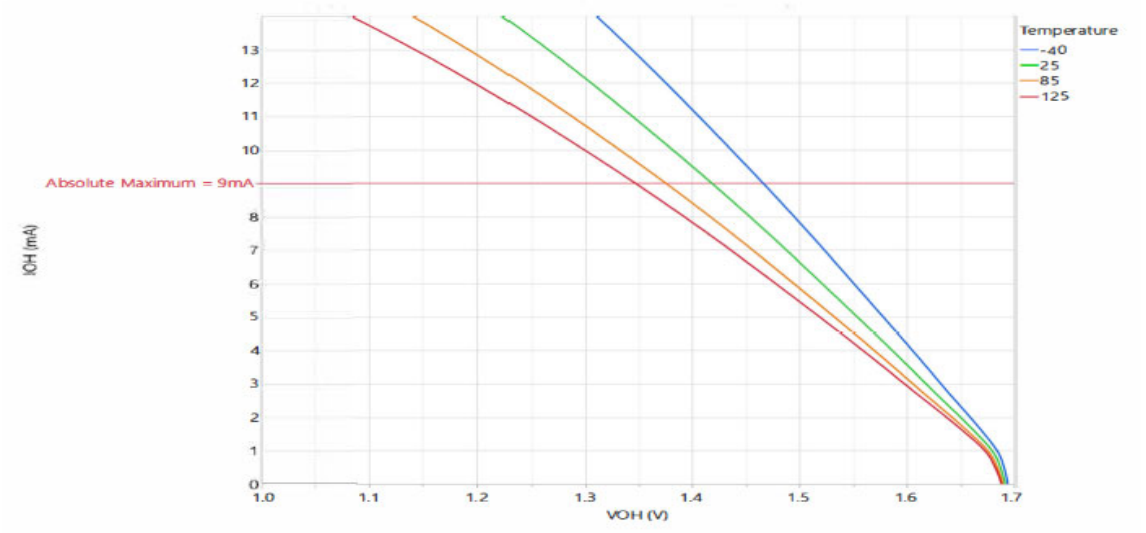
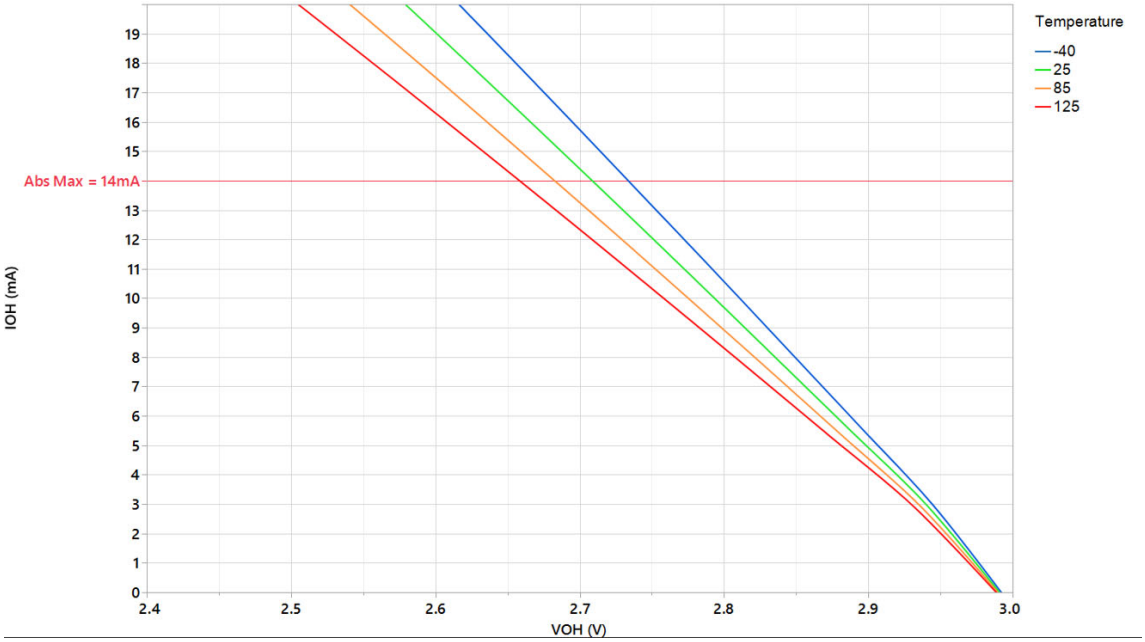



**Figure 40-14.** IOH vs VOH, 8x Drive Strength I/O on Typical parts with VDDIO = 3.0 V




## 40.13. Internal Voltage Reference Electrical Specifications (IVREF)

**Table 40-18.** Internal Voltage Reference DC Electrical Specifications

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. (1) | Max. | Units | Conditions |
| VR_1 | IVREF | ADC Internal Analog Channel Voltage | 1.168 | 1.2 | 1.223 | V | ADC.CTRLB.ADCHSEL = 1.2V IVREF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-18.** Internal Voltage Reference DC Electrical Specifications (continued)

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. (1) | Max. | Units | Conditions |
| VR_9 | ACIREF(2) | Internal Comparator Voltage Reference | 0.787 | 0.8024 | 0.84 | V | AC.COMPCTRLn.MUXNEG = Bandgap |
| VR_25 | TDRIFT | Internal Voltage Reference Temperature Drift | — | — | 0.34 | %/°C | Over full operating temperature range |
| VR_27 | VDRIFT | Internal Voltage Reference Voltage Drift | — | — | 2.328 | %/V | Over full operating voltage range |

**Notes:**

1. Typical values at 25°C.
2. See parameter CMP_23 for Comparator Reference voltage range information.

## 40.14. Maximum Clock Frequencies

**Table 40-19.** Maximum Clock Frequencies AC Electrical Specifications

| AC CHARACTERISTICS | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | |
|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Max | Units |
| FCLK_1 | FCY | MCU clock freq | 72 | MHz |
| FCLK_3 | fAHB | AHB clock freq | 72 | MHz |
| FCLK_5 | fAPB0:1] | APB clock freq | 72 | MHz |
| FCLK_5 | fAPB2 | | 53 | MHz |
| FCLK_6 | fGCLKGEN[0] | GCLK clock freq output | 200 | MHz |
| | fGCLKGEN[1:7] | | 200 | MHz |
| | fGCLKGEN[8:11] | | 100 | MHz |
| FCLK_7 | fGCLK_PLL | PLL reference clock freq | 60 | MHz |
| FCLK_11 | fGCLK_DFLL48M_REF | DFLL 48M reference clock freq | 1 | MHz |
| FCLK_13 | fGCLK_EIC | EIC input clock freq | 100 | MHz |
| FCLK_15 | fGCLK_FREQM_MSR | FREQM Measure Clock Frequency | 200 | MHz |
| FCLK_17 | fGCLK_FREQM_REF | FREQM Reference Clock Frequency | 100 | MHz |
| FCLK_19 | fGCLK_EVSYS_CHANNELx | EVSYS channel x input clock freq | 100 | MHz |
| FCLK_21 | fGCLK_SERCOMx_SLOW | Common SERCOM slow input clock freq | 32 | kHz |
| FCLK_23 | fGCLK_SERCOM_CORE | SERCOM input clock freq | 160 | MHz |
| FCLK_25 | fGCLK_CANx | CAN input clock freq | 100 | MHz |
| FCLK_27 | fGCLK_USBFS | USB FS input clock freq | 48 | MHz |
| FCLK_35 | fGCLK_TCCx | TCCx input clock freq | 200 | MHz |
| FCLK_43 | fGCLK_CCL | CCL input clock freq | 50 | MHz |
| FCLK_45 | fGCLK_GCLKINx | External GCLKx input clock freq | 50 | MHz |
| FCLK_49 | fGCLK_AC | Analog comparator peripheral module clock freq | 100 | MHz |
| FCLK_51 | fGCLK_ADCx | ADCx input clock freq | 144 | MHz |
| FCLK_55 | fGCLK_PTC | PTC input clock freq | 50 | MHz |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.15. XOSC Electrical Characteristics

**Table 40-20.** External XTAL and Clock AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions (1) |
| XOSC_1 | FOSC_XOSC | XOSC Crystal Frequency | 4 | — | 48 | MHz | XOSCCTRLn.XTALEN = 1<br>XIN, XOUT Primary Osc |
| XOSC_1A | TOSC | TOSC = 1/ FOSC_XOSC | 20.83 | — | 250 | ns | See parameter XOSC1 for FOSC_XOSC value |
| XOSC_2 | XOSC_ST (1) | XOSC Crystal Start-up Time | — | — | 1300000 (3) | TOSC | Crystal stabilization time only, not Oscillator Ready<br>XOSCCTRLA.AGC = 1<br>FOSC = 48 MHz<br>**(Note 1)** |
| XOSC_3 | CXIN | XOSC XIN parasitic pin capacitance | — | 2.2 | — | pF | — |
| XOSC_5 | CXOUT | XOSC XOUT parasitic pin capacitance | — | 2.2 | — | pF | — |
| XOSC_11 | CLOAD (2) | Crystal load capacitance FOSC_XOSC = 4 MHz | — | — | 20 | pF | XOSCCTRLA.AGC = 1<br>Crystal ESR ≤ 500Ω |
| XOSC_13 | | Crystal load capacitance FOSC_XOSC = 8 MHz | — | — | | pF | XOSCCTRLA.AGC = 1<br>Crystal ESR ≤ 500Ω |
| XOSC_15 | | Crystal load capacitance FOSC_XOSC = 12 MHz | — | — | | pF | XOSCCTRLA.AGC = 1<br>Crystal ESR ≤ 230Ω |
| XOSC_17 | | Crystal load capacitance FOSC_XOSC = 16 MHz | — | — | 18 | pF | XOSCCTRLA.AGC = 1<br>Crystal ESR ≤ 160Ω |
| XOSC_19 | | Crystal load capacitance FOSC_XOSC = 24 MHz | — | — | | pF | XOSCCTRLA.AGC = 1<br>Crystal ESR ≤ 70Ω |
| XOSC_21 | | Crystal load capacitance FOSC_XOSC = 32 MHz | — | — | 12 | pF | XOSCCTRLA.AGC = 1<br>Crystal ESR ≤ 80Ω |
| XOSC_23 | | Crystal load capacitance FOSC_XOSC = 48 MHz | — | — | 8 | pF | XOSCCTRLA.AGC = 1<br>Crystal ESR ≤ 70Ω |
| XOSC_33 | DLEVEL | MCU Crystal Osc Power Drive Level | — | — | 100 | µW | XOSCCTRLA.AGC = 1 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-20.** External XTAL and Clock AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions (1) |
| XOSC_35 | FOSC_XCLK | Ext Clock Oscillator Input Freq (XIN pin) | 4 | — | 48 | MHz | XOSCCTRLA.XTALEN = 0 |
| XOSC_37 | XCLK_DC | Ext Clock Oscillator (XIN) Duty Cycle | 40 | 50 | 60 | % | XOSCCTRLA.XTALEN = 0 |
| XOSC_39 | XCLK_FST | Primary XIN Clock Fail Safe Time-out Period | — | 4*1/ (DFLL_1/2^XOSCCTRLA.CFDPRESC) | — | µs | — |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-20.** External XTAL and Clock AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions (1) |

**Notes:**

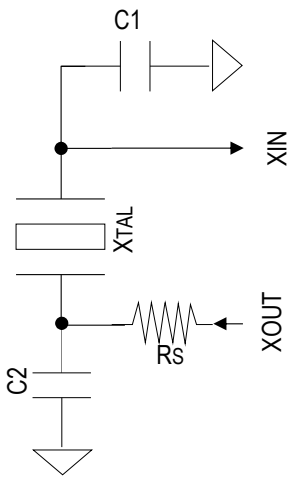
1. This is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern, the customer would need to characterize this based on their design choices.
2. The Crystal Load Capacitor calculation is as follows:
   - Standard PCB trace capacitance = 1.5 pF per 12.5 mm (0.5 inches) (i.e., PCB STD TRACE W = 0.175 mm, H = 36 µm, T = 113 µm)
   - XTAL PCB capacitance typical therefore ~= 2.5 pF for a tight PCB XTAL layout
   - For $C_{XIN}$ and $C_{XOUT}$ within 4 pF of each other, Assume CXTAL_EFF = (($C_{XIN}$+$C_{XOUT}$) / 2)
     **Note:** Averaging $C_{XIN}$ and $C_{XOUT}$ will effect final calculated $C_{LOAD}$ value by less than 0.25 pF.

     **Notes:**
     **Equation 1:**
     MFG $C_{LOAD}$ Spec = {( [$C_{XIN}$ + C1] * [$C_{XOUT}$ + C2] ) / [$C_{XIN}$ + C1 + C2 + $C_{XOUT}$] } + estimated oscillator PCB stray capacitance
     - Assuming C1 = C2 and $C_{XIN}$ ~= $C_{XOUT}$, the formula can be further simplified and restated to solve for C1 and C2 by:

     **Equation 2 (Simplified version of equation 1):**
     C1 = C2 = ((2 * MFG $C_{LOAD}$ spec) - $C_{XTAL_EFF}$ - (2 * PCB capacitance))
     **For example:**
     - XTAL Mfg $C_{LOAD}$ Data Sheet Spec = 12 pF
     - PCB XTAL trace Capacitance = 2.5 pF
     - $C_{XIN}$ pin = 6.5 pF, $C_{XOUT}$ pin = 4.5 pF, therefore $C_{XTAL_EFF}$ = (($C_{XIN}$+$C_{XOUT}$) / 2)

     $C_{XTAL_EFF}$ = ((6.5 + 4.5)/2) = 5.5 pF
     C1 = C2 = ((2 * MFG $C_{LOAD}$ spec) - $C_{XTAL_EFF}$ - (2 * PCB capacitance))
     C1 = C2 = (24 - 5.5 - (2 * 2.5))
     C1 = C2 = 13.5 pF (Always rounded down)
     C1 = C2 = 13 pF (i.e., for hypothetical example crystal external load capacitors)
     User C1 = C2 = 13 pF ≤ $C_{LOAD}$(max) spec

     **Figure 40-15.** XTAL




3. Maximum start up time; user must configure XOSCCTRLA.STARTUP counter value greater than start up time.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.16. XOSC32K Electrical Characteristics

**Table 40-21.** XOSC32K AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| XOSC32_1 | FOSC_XOSC32 | XOSC32 Oscillator Crystal Frequency | — | 32.768 | — | kHz | XIN32, XOUT32 Secondary Osc |
| XOSC32_3 | CXIN32 | XOSC32 XIN32 parasitic pin capacitance | — | 4.9 | — | pF | — |
| XOSC32_5 | CXOUT32 | XOSC32 XOUT32 parasitic pin capacitance | — | 4.8 | — | pF | — |
| XOSC32_11 | CLOAD_X32 (2) | 32.768 kHz Crystal Load Capacitance | — | — | 12.5 | pF | XOSC32K.CGM ≥ 6<br>XOSC32K.BOOST = 0<br>XOSC32K.XTALEN = 1<br>XOSC32K.ENABLE = 1 |
| XOSC32_13 | ESR_X32 | 32.768 kHz Crystal ESR | — | — | 60 | KΩ | XOSC32K.CGM ≥ 6<br>XOSC32K.BOOST = 0<br>XOSC32K.XTALEN = 1<br>XOSC32K.ENABLE = 1<br>Cload = 12.5 pF |
| XOSC32_14 | | | — | — | 100 | KΩ | XOSC32K.CGM ≥ 11<br>XOSC32K.BOOST = 0<br>XOSC32K.XTALEN = 1<br>XOSC32K.ENABLE = 1<br>Cload = 12.5 pF |
| XOSC32_15 | TOSC32 | TOSC32 = 1/ FOSC_XOSC32 | — | 30.5176 | — | µs | See parameter XOSC32_1 for FOSC_XOSC32 value |
| XOSC32_17 | XOSC32_ST (1) | XOSC32 Crystal Start-up Time | — | — | 14000[3] | TOSC | XOSC32K.CGM ≥ 6<br>XOSC32K.BOOST = 0<br>XOSC32K.XTALEN = 1<br>XOSC32K.ENABLE = 1<br>Crystal ESR = 100 KΩ<br>Cload = 12.5 pF (1)<br>Crystal stabilization time only not Oscillator Ready |
| XOSC32_19 | FOSC_XCLK32 | Ext Clock Oscillator Input Freq (XIN32 pin) | 31.13 | 32.768 | 34.406 | kHz | XOSC32K.XTALEN = 0<br>XOSC32K.ENABLE = 1 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-21.** XOSC32K AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| XOSC32_21 | XCLK32_DC | Ext Clock Oscillator Duty Cycle | 25 | 50 | 75 | % | XOSC32K.XTALEN = 0<br>XOSC32K.ENABLE = 1 |
| XOSC32_23 | XCLK32_FST | XIN32 Clock Fail Safe Time-out Period | — | 4*1/ (LP32K_1/2^CFDCTRL.CFDPRESC) | — | µs | — |

**Notes:**

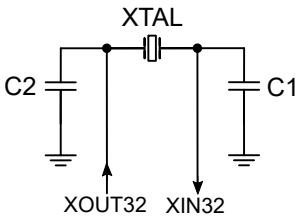
1. This is for guidance only. A major component of crystal start-up time is based on the second party crystal MFG parasitics that are outside the scope of this specification. If this is a major concern the customer would need to characterize this based on their design choices.
2. The Crystal Load Capacitor calculation is as follows:
   - Standard PCB trace capacitance = 1.5 pF per 12.5 mm (0.5 inches) (i.e., PCB STD TRACE W = 0.175 mm, H = 36 µm, T = 113 µm)
   - XTAL PCB capacitance typical, therefore ~= 2.5 pF for a tight PCB XTAL layout
   - For $C_{XIN32}$ and $C_{XOUT32}$ within 4 pF of each other, assume CXTAL_EFF = ((CXIN32+CXOUT32) / 2)
     **Note:** Averaging $C_{XIN32}$ and $C_{XOUT32}$ will effect final calculated $C_{LOAD}$ value by less than 0.25 pF.

     **Equation 1:**

     MFG $C_{LOAD}$ Spec = {( [$C_{XIN32}$ + C1] * [$C_{XOUT32}$ + C2] ) / [$C_{XIN32}$ + C1 + C2 + $C_{XOUT32}$] } + estimated oscillator PCB stray capacitance
     - Assuming C1 = C2 and $C_{XIN32}$ ~= $C_{XOUT32}$, the formula can be further simplified and restated to solve for C1 and C2 by:

     **Equation 2 (Simplified Equation 1):**

     C1 = C2 = ((2 * MFG $C_{LOAD}$ spec) - $C_{XTAL_EFF}$ - (2 * PCB capacitance))

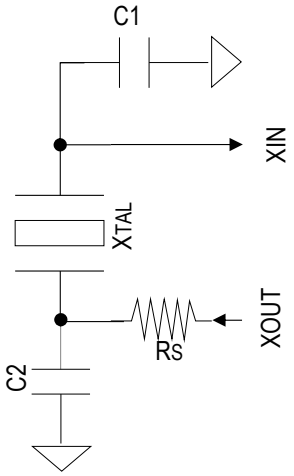
     
     

     **For example:**
     - XTAL Mfg $C_{LOAD}$ Data Sheet Spec = 12 pF
     - PCB XTAL Trace Capacitance = 2.5 pF
     - $C_{XIN32}$ pin = 6.5 pF, $C_{XOUT32}$ pin = 4.5 pF. Therefore, $C_{XTAL_EFF}$ = (($C_{XIN32}$+$C_{XOUT32}$) / 2)

     $C_{XTAL_EFF}$ = ((6.5 + 4.5)/2) = 5.5 pF
     C1 = C2 = ((2 * MFG $C_{LOAD}$ spec) - $C_{XTAL_EFF}$ - (2 * PCB capacitance))
     C1 = C2 = (24 - 5.5 - (2 * 2.5))
     C1 = C2 = (24 - 5.5 - 5)
     C1 = C2 = 13.5 pF (always rounded down)
     C1 = C2 = 13 pF (i.e., for hypothetical example crystal external load capacitors)
     User C1 = C2 = 13 pF ≤ $C_{LOAD}$_X32(max) spec
3. User Selectable in XOSC32K.STARTUP.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-16.** XTAL




## 40.17. OSCULP32K Electrical Characteristics

**Table 40-22.** Low-Power Internal 32 kHz RC Oscillator AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| LP32K_1 | FOSC_LPRC32K | Output Frequency | 30.1 | 32.768 | 36.8 | kHz | -40°C ≤ TA ≤ +85°C (i.e., Factory Default Calibration) |
| LP32K_9 | RC32K_Duty | LPRC32K OSC Duty Cycle | — | 50 | — | % | |

## 40.18. DFLL-FDPLL Electrical Characteristics

**Table 40-23.** DFLL AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| **DFLL48 MHz (Open Loop) (1,2)** | | | | | | | |
| DFLL_1 | DFLL_OL_FOUT | DFLL Open Loop Clock Frequency | 45.600 | 48.000 | 49.120 | MHz | Normal mode (DFLLCTRLA.LOWFREQ = 0) |
| | | | 6.760 | 8.000 | 9.200 | | Low-Frequency mode (DFLLCTRLA.LOWFREQ = 1) |
| DFLL_9 | DFLL_OL_SRT | Start-Up (Ready bit valid) | — | 13 | 16 | µs | Normal mode (DFLLCTRLA.LOWFREQ = 0) |
| | | | — | 68 | 75 | | Low-Frequency mode (DFLLCTRLA.LOWFREQ=1) |
| **DFLL48 MHz (Closed Loop) (3,4)** | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-23.** DFLL AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| DFLL_11 | DFLL_CL_FIN | DFLL Closed Loop Input Frequency Range (4) | 1000 | 32768 | 1000000 | Hz | — |
| DFLL_13 | DFLL_CL_FOUT | DFLL Closed Loop Clock Frequency (6) | 47.88 | 48.00 | 48.12 | MHz | Normal mode (DFLLCTRLA.LOWFREQ = 0)<br>XOSC32 32.768 kHz PPM ≤ 100,<br>DFLLMUL = 1465 |
| DFLL_14 | | | 7.98 | 8.00 | 8.02 | | Low-Frequency mode (DFLLCTRLA.LOWFREQ = 1)<br>XOSC32 32.768 kHz PPM ≤ 100,<br>DFLLMUL = 244 |
| DFLL_15 | DFLL_CL_Jitter | DFLL Period Jitter Pk-to-Pk | — | 0.920 | 2.700 | % | Normal mode (DFLLCTRLA.LOWFREQ = 0)<br>XOSC32 32.768 kHz PPM ≤ 100,<br>DFLLMUL = 1465 |
| DFLL_17 | | | — | 1.000 | 2.000 | % | Low- Frequency mode (DFLLCTRLA.LOWFREQ = 1)<br>XOSC32 32.768 kHz PPM ≤ 100,<br>DFLLMUL = 244 |
| DFLL_21 | DFLL_CL_SRT | DFLL Closed- Loop mode / Lock Time (5, 6) | — | 0.460 | 1.100 | ms | Normal mode (DFLLCTRLA.LOWFREQ = 0)<br>XOSC32 32.768 kHz PPM ≤ 100,<br>DFLLMUL = 1465 |
| DFLL_23 | | | — | 0.350 | 1.000 | | Low-Frequency mode (DFLLCTRLA.LOWFREQ = 1)<br>XOSC32 32.768 kHz PPM ≤ 100,<br>DFLLMUL = 244 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

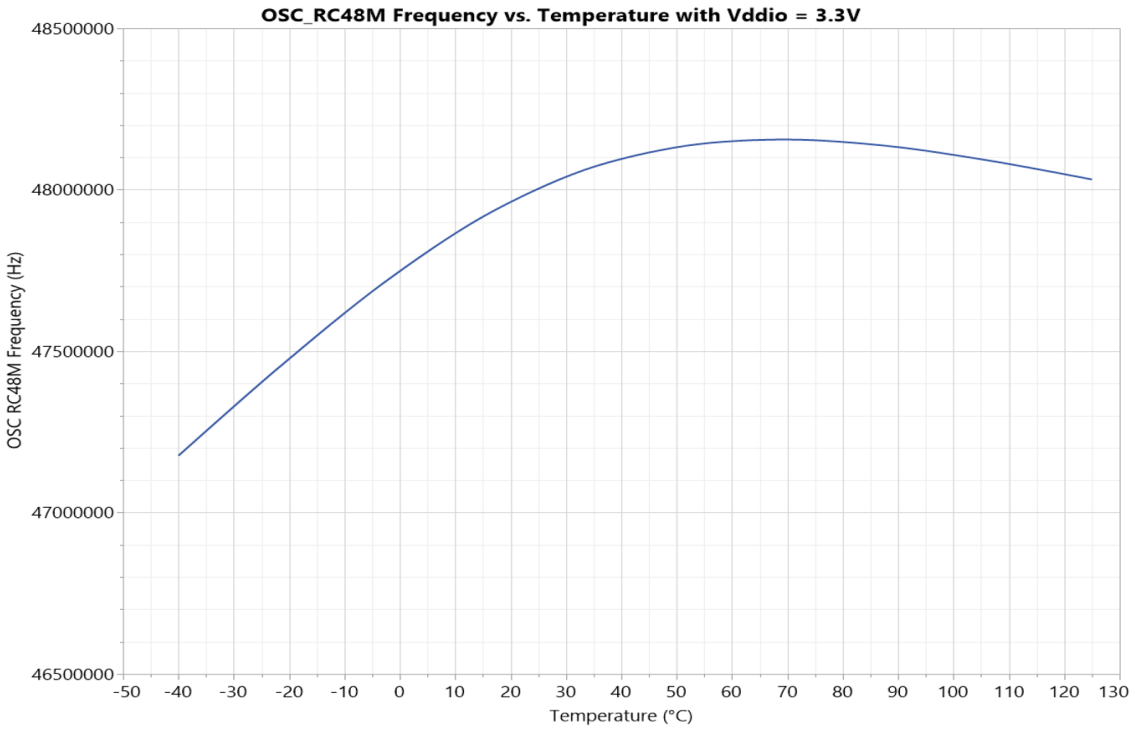



**Table 40-24.** PLL (Frequency Digital Phase Locked Loop) AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| **PLLxxMHz (Digital Phase-Locked Loop)** | | | | | | | |
| PLL_1 | PLL_FIN | PLL Input Frequency Range | 4 | — | 48 | MHz | — |
| PLL_3 | PLL_FOUT | PLL Output Clock Frequency | 12.7 | — | 200 | MHz | |
| PLL_11 | PLL_SRT | Lock Time (7) | — | 25 | — | µs | — |

**Notes:**

1. In Open-Loop mode, the DFLL uses a self contained internal RC oscillator clock source. its course calibrated value is loaded at out of reset. In addition, there is a fine tune trim register (DFLLTUNE) the user software can assess.
2. Not recommended for functional USB operation, SOF sync start-up only.
3. In Closed-Loop mode the DFLL can use a varity of clock sources. The DFLL can be trimmed using the DFLLMUL register.
4. To ensure that the DFLL stays within the +/-0.25% of its clock frequency, any reference clock for DFLL in close loop must be within a 8% maximum error accuracy.
5. REFCLK for DFLL or XOSC32K,PLL is XOSC.
6. DFLLCTRLB.QLDIS = 0: quick lock enabled (default), DFLLCTRLB.CCDIS = 0: chill cycles enabled (default).
   DFLLMUL.STEP = 8: Maximum fine step size, divided or dividing into two parts, search 8 is optimum value.
   During a maximum of 30 cycles of the reference clock period (250 cycles for 1Mhz), between lock flag asserted and frequency stabilization, DFLL frequency accuracy will be limited to +/-1.5% at 48Mhz (+/-3% at 8Mhz); after this duration, the frequency accuracy is within +/-0.25%.
   At 1khz reference clock and to maintain the frequency accuracy within +/-0.25% for 48Mhz, a reduced STEP value at 4 (instead of the optimum 8) eliminates this period of inaccuracy, at the expense of maximum 15% of lock time.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 40.19. Analog-to-Digital Converter (ADC) Electrical Specifications

**Table 40-25.** ADC AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| **Device Supply** | | | | | | | |
| ADC_1 | AVDD | ADC Module Supply | AVDD(min) | — | AVDD(max) | V | VDDIOx = AVDD |
| **Reference Inputs** | | | | | | | |
| ADC_3 | VREF(1) | ADC Reference Voltage | 2.4 | — | AVDD | V | VREF ≤ AVDD |
| **Analog Input Range** | | | | | | | |
| ADC_7 | AFS | Full-Scale Analog Input Signal Range | AVSS | — | VREF | V | Single-Ended mode |
| | | | -VREF | — | +VREF | V | Differential mode, VCMIN = VREF/2 |
| ADC_9 | VCMIN | Input common mode voltage | — | VREF/2 | — | V | — |

**Note:**

1. ADC functional device operation with either internal or external VREF<2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately ~((0.06 * 2n) / VREF) LSB's over full scale range, where "n"=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus users application noise/accuracy on AVDD, AVSS.

**Table 40-26.** ADC Single Ended Mode AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| **Single Ended Mode ADC Accuracy(4)** | | | | | | | |
| SADC_11 | Res | Resolution | 8 | — | 12 | bits | Selectable 8, 10, 12 bit Resolution Ranges |
| SADC_13 | INL (3) | Integral Nonlinearity | -2.2 | -1.7/1.1 | 1.3 | LSB | 4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V |
| SADC_19 | DNL (3) | Differential Nonlinearity | -1.4 | -1/1.8 | 2.4 | LSB | 4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V |
| SADC_25 | GERR (3,6) | Gain Error | -4.7 | — | -1.2 | LSB | 4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V |
| SADC_31 | EOFF (3,6) | Offset Error | 1.9 | — | 5.1 | LSB | 4.5 Msps, Internal VREF = AVDD = VDDIO = 3.3V |
| **Single Ended Mode ADC Dynamic Performance (1,2,4)** | | | | | | | |
| SADC_43 | ENOB (3) | Effective Number of bits | 10.3 | 10.75 | — | bits | VREF = AVDD = VDDIO = 3.3V @ 12bit at 4.5Msps<br>Slew rate control enabled on IOs (slewlim > 0) |
| SADC_45 | SINAD (1,2,3) | Signal-to-Noise and Distortion | 63 | 66 | — | dB | |
| SADC_47 | SNR (1,2,3) | Signal-to-Noise ratio | 64 | 66 | — | | |
| SADC_51 | THD (1,2,3,5) | Total Harmonic Distortion | — | -81 | -77 | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-26.** ADC Single Ended Mode AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |

**Notes:**

1. Characterized with an analog input sine wave = (FTP(max) / 100). Example: FTP(max) = 1Msps / 100 = 10kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12bit resolution.
3. ADC is configured in 12bits mode, All registers are at the reset default value unless otherwise stated, and XOSC48M AGC is ON (i.e. XOSCCTRLA.AGC=0x1).
4. ADC functional device operation with either internal or external VREF < 2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately ~((0.06 * 2n) / VREF) LSB's over full scale range, where "n"=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus users application noise/accuracy on VDDANA, GNDANA.
5. Value taken over 7 harmonics.

**Table 40-27.** ADC Differential Mode AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| **Differential Mode ADC Accuracy(4)** | | | | | | | |
| DADC_11 | Res | Resolution | 8 | — | 12 | bits | Selectable 8, 10, 12 bit Resolution Ranges |
| DADC_13 | INL (3) | Integral Nonlinearity | -2.3 | -1.6/1.5 | 2.2 | LSB | 4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V |
| DADC_19 | DNL (3) | Differential Nonlinearity | -1 | -1/1.9 | 2.7 | LSB | 4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V |
| DADC_25 | GERR (3,6) | Gain Error | -4.3 | — | 0 | LSB | 4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V |
| DADC_31 | EOFF (3,6) | Offset Error | -0.1 | — | 1.9 | LSB | 4.5 Msps, Internal VREF = AVDDA = VDDIO = 3.3V |
| **Differential Mode ADC Dynamic Performance (1,2,4)** | | | | | | | |
| DADC_43 | ENOB (3) | Effective Number of bits | 10.5 | 11.25 | — | bits | VREF = AVDD = VDDIO = 3.3V at 12 bit at 4.5 Msps |
| DADC_45 | SINAD (1, 2, 3) | Signal-to-Noise and Distortion | 65 | 69 | — | dB | |
| DADC_47 | SNR (1, 2, 3) | Signal-to-Noise ratio | 65 | 69 | — | | |
| DADC_51 | THD (1, 2, 3, 5) | Total Harmonic Distortion | — | -86 | -80 | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-27.** ADC Differential Mode AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |

**Notes:**

1. Characterized with an analog input sine wave = (FTP(max) / 100). Example: FTP(max) = 1Msps / 100 = 10kHz sine wave.
2. Sine wave peak amplitude = 96% ADC_ Full Scale amplitude input with 12bit resolution.
3. ADC is configured in 12bits mode, All registers are at the reset default value unless otherwise stated, and XOSC48M AGC is ON (i.e. XOSCCTRLA.AGC=0x1).
4. ADC functional device operation with either internal or external VREF < 2.4V is functional, but not characterized. ADC will function, but with degraded accuracy of approximately ~((0.06 * 2n) / VREF) LSB's over full scale range, where "n"=#bits. ADC accuracy is limited by internal VREF accuracy + drift, MCU generated noise plus users application noise/accuracy on VDDANA, GNDANA.
5. Value taken over 7 harmonics.

**Table 40-28.** ADC Conversion and Sample AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| **ADC Clock Requirements** | | | | | | | |
| ADC_53 | TAD | ADC Clock Period | 13.88 | — | 7145 | ns | VREF = AVDD = 3.3V and Res = 6,8,10 bit |
| | | | 13.88 | — | 1250 | ns | VREF = AVDD = 3.3V and Res = 12 bit |
| ADC_55 | fGCLK_ADCx | ADCx Module GCLK max input freq | — | — | FCLK_51 | MHz | VREF = AVDD = 3.3V |
| **ADC Throughput Rates** | | | | | | | |
| ADC_57 | FTPR (1) | Sample-Rate for ADC0 with SAMC = 0 (min) | — | — | 4.5 | Msps | 12-bit resolution, Rsource Impedance ≤ 200 Ω |
| | | | — | — | 5.14 | | 10-bit resolution, Rsource Impedance ≤ 350 Ω |
| | | | — | — | 6 | | 8-bit resolution, Rsource Impedance ≤ 500 Ω |
| | | | — | — | 7.2 | | 6-bit resolution, Rsource Impedance ≤ 650 Ω |
| **ADC Conversion and Sample time** | | | | | | | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-28.** ADC Conversion and Sample AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| ADC_59 | TSAMP | Sample-Time for ADC0 | 3 | — | — | TAD | 12 bit TAD(min), Ext Analog Input Rsource ≤ 200 Ω, Max ADC Clock |
| | | | | | | | 10 bit TAD(min), Ext Analog Input Rsource ≤ 350 Ω, Max ADC Clock |
| | | | 8 | — | — | | 12 bit TAD(min), Ext Analog Input Rsource ≤ 500 Ω, Max ADC Clock |
| | | | | | | | 10 bit TAD(min), Ext Analog Input Rsource ≤ 700 Ω, Max ADC Clock |
| | | | 14 | — | — | | 12 bit TAD(min), Ext Analog Input Rsource ≤ 1 kΩ, Max ADC Clock |
| | | | | | | | 10 bit TAD(min), Ext Analog Input Rsource ≤ 1.25 kΩ, Max ADC Clock |
| | | | 64 | — | — | | 12 bit TAD(min), Ext Analog Input Rsource ≤ 5 kΩ, Max ADC Clock |
| | | | | | | | 10 bit TAD(min), Ext Analog Input Rsource ≤ 5.5 kΩ, Max ADC Clock |
| ADC_61 | TCNV | Conversion Time (after sample time is complete) | 13 | | | TAD | 12-bit resolution |
| | | | 11 | | | | 10-bit resolution |
| | | | 9 | | | | 8-bit resolution |
| | | | 7 | | | | 6-bit resolution |
| ADC_63 | Twarm-up | Warm Up Time after CTRLA.ANAEN = 1 and CTRLA.ENABLE = 1 | — | — | 500 TAD or 20 µs, which ever is bigger | µs | — |

**Notes:**

1. ADC Throughput Rate FTP = ((1 / ((TSAMP + TCNV) * TAD)) / (# of user active analog inputs in use on specific target ADC module)).
2. Specification values assume only one AINx channel in use.

## 40.20. Analog Comparator (AC) Electrical Characteristics

**Table 40-29.** Comparator AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ.(1) | Max. | Units | Conditions |
| CMP_1 | VIOFF | Input Offset Voltage | -8 | — | 8 | mV | COMPCTRLn.HYST = 0x3.<br>Comparator ref voltage = AVDD/2 |
| CMP_4 | VIN | Input Voltage Range | AVSS | — | AVDD | V | With respect to GND and AVDD |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-29.** Comparator AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ.(1) | Max. | Units | Conditions |
| CMP_5 | VHYST_00 | Input Hysteresis Voltage | 2 | — | 14 | mV | COMPCTRLn.HYST = 0x0.<br>Comparator ref voltage = AVDD/2 |
| | VHYST_01 | Input Hysteresis Voltage | 6 | — | 26 | mV | COMPCTRLn.HYST = 0x1.<br>Comparator ref voltage = AVDD/2 |
| | VHYST_02 | Input Hysteresis Voltage | 20 | — | 56 | mV | COMPCTRLn.HYST = 0x2.<br>Comparator ref voltage = AVDD/2 |
| | VHYST_03 | Input Hysteresis Voltage | 32 | — | 86 | mV | COMPCTRLn.HYST = 0x3.<br>Comparator ref voltage = AVDD/2 |
| CMP_15 | TRESPSSain<0:1> | Small Signal Response Time - High speed | — | — | 126 | ns | COMPCTRLn.SPEED = 0x0<br>COMPCTRLn.MUXPOS = 0x0<br>COMPCTRLn.MUXNEG = 0x1<br>COMPCTRLn.HYST = 0x3<br>Comparator ref voltage = AVDD/2.<br>Input overdrive = +/- 180 mV |
| | | Small Signal Response Time - Low Speed | — | — | 432 | ns | COMPCTRLn.SPEED = 0x1<br>COMPCTRLn.MUXPOS = 0x0<br>COMPCTRLn.MUXNEG = 0x1<br>COMPCTRLn.HYST = 0x3<br>Comparator ref voltage = AVDD/2.<br>Input overdrive = +/- 180 mV |
| CMP_19 | COUTVAL | Comparator Enabled to Output Valid - High Speed | — | — | 2 | us | Comparator module is configured before enabling it<br>COMPCTRLn.SPEED = 0x0 |
| | | Comparator Enabled to Output Valid - Low Speed | — | — | 20 | us | Comparator module is configured before enabling it<br>COMPCTRLn.SPEED = 0x1 |
| CMP_23 | CVREFRNG | Comparator Voltage Reference Input Range | — | — | — | V | — |
| CMP_25 | FGCLK_AC | Analog comparator peripheral module clock freq | — | — | FCLK_49 | MHz | See FCLK_49 in Maximum Clock Frequencies |

**Note:**

1. Comparator Ref voltage cannot exceed: VIN(max) - VIOFF(max) - CMP_5(max) - 100 mV) ≥CMP_23 ≥ (VIN(min) + abs(VIOFF(min)) + CMP_5(max) + 100 mV).

**Table 40-30.** DAC Module AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| DAC_1 | DRES | DAC Resolution | — | — | 7 | Bits | — |
| DAC_7 | VOUT | Output Voltage Range | AVSS | — | AVDD - 1LSB | V | — |
| DAC_17 | Tset_FS | DAC Full Scale Settling Time | — | — | 1.5 | µs | +/-4 LSB of final value for step size from 10% to 90% |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-30.** DAC Module AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.6V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| DAC_18a | Thold | DAC Holding Time | 0.29 | — | — | ms | This parameter applies only for Low-Power mode:<br>DACCTRLn.SHENm = 0x1<br>Time for 1 LSB voltage loss of the DAC output |
| DAC_18b | | | 0.65 | — | — | ms | This parameter applies only for Low-Power mode:<br>DACCTRLn.SHENm = 0x1<br>Time for 10 LSB voltage loss of the DAC output |
| SDAC_19 | INL | Integral Non Linearity | -0.4 | — | 0.4 | LSB | — |
| SDAC_21 | DNL | Differential Non Linearity | -0.3 | — | 0.4 | LSB | — |
| SDAC_23 | GERR | Gain Error | -1.2 | — | -0.2 | LSB | — |
| SDAC_25 | EOFF | Offset Error | -0.7 | — | -0.7 | LSB | Normal mode:<br>DACCTRLn.SHENm = 0x0 |
| | | | -5.7 | — | 5.7 | LSB | Low-Power mode:<br>DACCTRLn.SHENm = 0x1 |

## 40.21. Peripheral Touch Controller (PTC) Electrical Characteristics

**Table 40-31.** Peripheral Touch Controller (PTC) AC Electrical Specifications

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | | Min. | Typ. | Max. | Units | Conditions |
| PTC_1 | CLOAD SC [1] | Self Capacitance mode (PTC Channel Y0 - Y35) | Y0 | — | — | 65 | pF | Maximum sensor load capacitance [1] |
| | | | Y1-35 | — | — | 70 | pF | |
| PTC_3 | CLOAD MC [1] | Mutual Capacitance mode (PTC Channel Y0 - Y35) | | — | — | 35 | pF | |
| PTC_4A | fPTC [2] | PTC frequency | | — | — | 4 | MHz | — |
| PTC_4B | Twarm | Warm-up time | | 500 PTC clock cycles | — | — | µs | — |

**Notes:**

1. Maximum capacitive load that the PTC circuitry can compensate for each channel.
2. fPTC = GCLK_PTC / PTC prescaler.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.22. Serial Peripheral Interface (SPI) Timings Electrical Characteristics

**Figure 40-17.** SPIx Host Module (CPHA = 0) Timing Diagrams

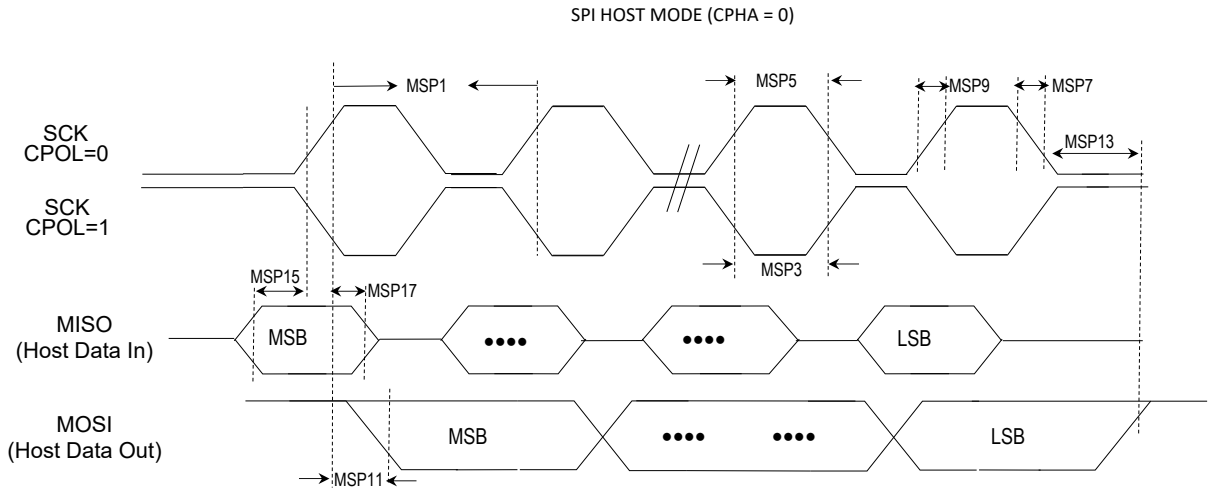
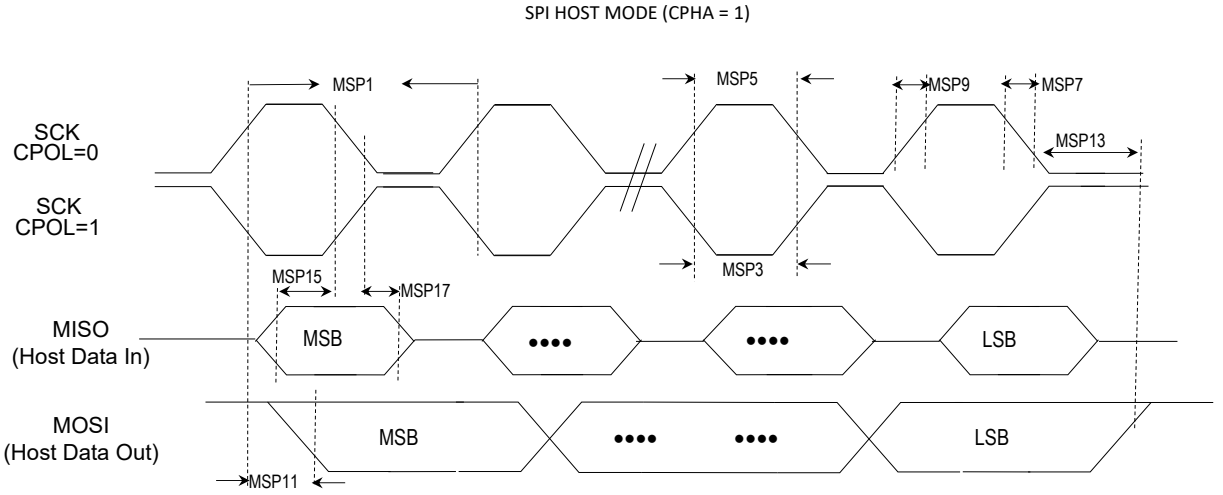



**Figure 40-18.** SPIx Host Module (CPHA = 1) Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-32.** SPIx Module Host Mode Electrical Specifications [1]

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics [1] | | Min. | Typ. | Max. | Units | Conditions |
| MSP_1 | FSCK | SCK Frequency | Normal Speed SERCOM | — | — | 22 | MHz | Transmitter mode, CTRB.RXEN = 0. CLOAD = 30 pF(max) |
| | | | | — | — | 22 | | Full Duplex Transmit and Receive mode, Loop-Back mode with one SPI talking to another SPI on the same MCU. CLOAD = 30 pF(max) |
| | | | | — | — | 1/(2*(TMIS+NOTE2_TV)) [2] | | Full Duplex Transmit and Receive mode, CLOAD = 30 pF (max).<br>The maximum SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the Loop-Back mode above may therefore be possible using the formula. |
| | | | High-Speed SERCOM [3] | — | — | 46 | MHz | Transmitter mode, CTRB.RXEN = 0, VDDIOx >= 2.97V,<br>CLOAD = 20 pF(max) |
| | | | | — | — | 1/(2*(TMIS_HS+NOTE2_TV)) [2] | | Full-Duplex Transmit and Receive mode, CLOAD=20 pF(max).<br>The maximum SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the Loop- Back mode above may therefore be possible using the formula.<br>VDDIOx >= 2.97V |
| MSP_3 | TSCL | SCK Output Low Time | | 1/(2*FSCK) | — | — | ns | — |
| MSP_5 | TSCH | SCK Output High Time | | 1/(2*FSCK) | — | — | ns | — |
| MSP_7 | TSCF | SCK and MOSI Output Fall Time | | — | — | DI_27 or DI_31 | ns | See parameter in I/O Specifications |
| MSP_9 | TSCR | SCK and MOSI Output Rise Time | | — | — | DI_25 or DI_29 | ns | See parameter in I/O Specifications |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

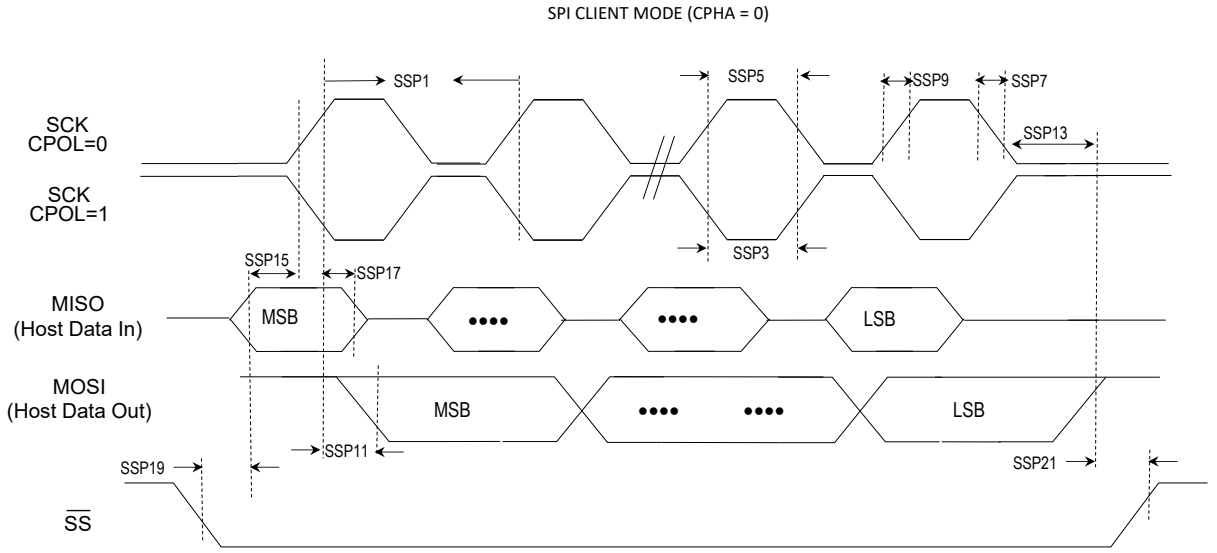
**Table 40-32.** SPIx Module Host Mode Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics [1] | | Min. | Typ. | Max. | Units | Conditions |
| MSP_11 | TMOV | MOSI Data Output Valid after SCK | Normal Speed SERCOM | — | — | 9.56 | ns | — |
| | | | High-Speed SERCOM [3] | — | — | 2.94 | ns | VDDIOx >= 2.97V |
| MSP_13 | TMOH | MOSI hold after SCK | | 0 | — | — | ns | — |
| MSP_15 | TMIS | MISO Setup Time of Data Input to SCK | Normal Speed SERCOM | 12.8 | — | — | ns | — |
| | | | High-Speed SERCOM [3] | 1.31 | — | — | ns | VDDIOx >= 2.97V |
| MSP_17 | TMIH | MISO Hold Time of Data Input to SCK | Normal Speed SERCOM | 0 | — | — | ns | — |
| | | | High Speed SERCOM [3] | 1.49 | — | — | ns | VDDIOx >= 2.97V |
| MSP_19 | SPI_GCLK | SERCOM SPI input clk freq, GCLK_SPI | | — | — | FCLK_23 | MHz | — |

**Notes:**

1. Assumes 30 pF external load on all SPIx pins unless otherwise noted.
2. NOTE2_TV is the client external device data output valid time from clock edge specification.
3. SERCOM0 pins PC0,PC1,PC2,PC3.

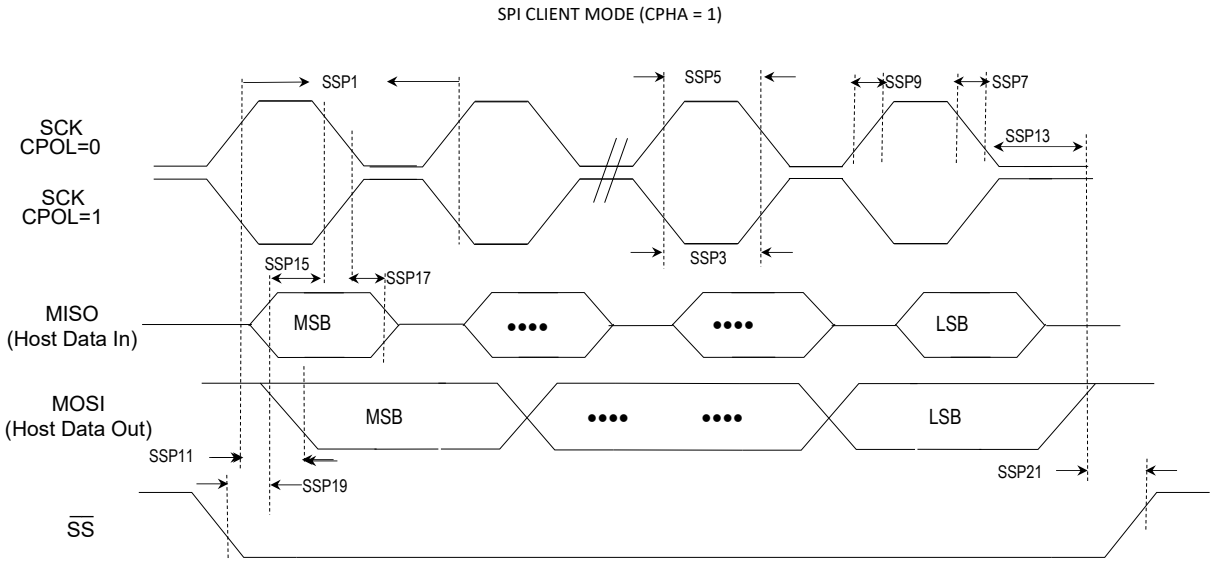
**Figure 40-19.** SPIx Client Module (CPHA = 0) Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-20.** SPIx Client Module (CPHA = 1) Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-33.** SPIx Module Client Mode Electrical Specifications

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics [1] | | Min. | Typical | Max. | Units | Conditions |
| SSP_1 | FSCK | SCK Frequency | Normal Speed SERCOM | — | — | 22 | MHz | Receiver mode. CLOAD = 30 pF(max) |
| | | | | — | — | 22 | | Full Duplex Transmit and Receive mode, Loop-Back mode with one SPI talking to another SPI on the same MCU.<br>CLOAD = 30 pF(max) |
| | | | | — | — | 1/(2*(TSOV+NOTE2_TMIS)) [2] | | Full Duplex Transmit and Receive mode, CLOAD=30 pF(max).<br>The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the Loop-Back mode above may therefore be possible using the formula. |
| | | | High-Speed SERCOM [3] | — | — | 46 | MHz | Receiver mode. VDDIOx >= 2.97V,<br>CLOAD = 20 pF(max) |
| | | | | — | — | 1/(2*(TSOV_HS+NOTE2_TMIS)) [2] | | Full Duplex Transmit and Receive mode, CLOAD=20 pF(max).<br>The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the loop back mode above may therefore be possible using the formula.<br>VDDIOx>=2.97V |
| SSP_3 | TSCL | SCK Output Low Time | | 1/(2*FSCK) | — | — | ns | — |
| SSP_5 | TSCH | SCK Output High Time | | 1/(2*FSCK) | — | — | ns | — |
| SSP_7 | TSCF | SCK and MOSI Output Fall Time | | — | — | DI_27 or DI_31 | ns | See parameter in I/O Specifications |
| SSP_9 | TSCR | SCK and MOSI Output Rise Time | | — | — | DI_25 or DI_29 | ns | See parameter in I/O Specifications |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-33.** SPIx Module Client Mode Electrical Specifications (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics (1) | | Min. | Typical | Max. | Units | Conditions |
| SSP_11A | TSOV | MISO Data Output Valid after SCK | Normal-Speed SERCOM | — | — | 14.83 | ns | — |
| | | | High-Speed SERCOM (3) | — | — | 10.52 | ns | VDDIOx>= 2.97V |
| SSP_13 | TSOH | MISO Hold after SCK | | 4 | — | — | ns | — |
| SSP_15 | TSIS | MOSI Setup Time of Data Input to SCK | Normal-Speed SERCOM | 7.33 | — | — | ns | — |
| | | | High-Speed SERCOM (3) | 2.22 | — | — | ns | VDDIOx>= 2.97V |
| SSP_17 | TSIH | MOSI Hold Time of Data Input to SCK | Normal Speed SERCOM | 2.24 | — | — | ns | — |
| | | | High-Speed SERCOM (3) | 1.92 | — | — | ns | VDDIOx>= 2.97V |
| SSP_19 | TSSS | $\overline{SS}$ setup to SCK (PRELOADEN = 0) | Normal Speed SERCOM | 5.76 | — | — | ns | — |
| | | | High-Speed SERCOM (3) | 1.47 | — | — | ns | VDDIOx>= 2.97V |
| SSP_21 | TSSH | $\overline{SS}$ hold after SCK Client | Normal Speed SERCOM | 0 | — | — | ns | — |
| | | | High-Speed SERCOM (3) | 0.36 | — | — | ns | VDDIOx>=2.97V |
| SSP_23 | SPI_GCLK | SERCOM SPI input clk freq, GCLK_SPI | — | — | — | FCLK_23 | MHz | — |

**Notes:**

1. Assumes 30 pF external load on all SPIx pins unless otherwise noted.
2. NOTE2_TMIS is the host external device setup time.
3. SERCOM0 pins PC0,PC1,PC2,PC3.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.23. Universal Asynchronous Receiver/Transmitter (UART) Electrical Characteristics

**Table 40-34.** UART AC Electrical Specifications

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | | Min. | Typ. | Max. | Units | Conditions |
| UT_1A | FBRATE | Baud Rate - Normal Speed SERCOM | Asynchronous SAMPR = 16x mode | — | — | 10 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 10 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_3A | | | Asynchronous SAMPR = 8x mode | — | — | 20 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 20 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_5A | | | Asynchronous SAMPR = 3x mode | — | — | 21 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 20 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_19A | | | Synchronous Mode X2 mode | — | — | 18 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 12 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_21A | | | Synchronous Mode X1 mode | — | — | 18 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 12 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_1B | FBRATE_HS | Baud Rate - High Speed SERCOM | Asynchronous SAMPR = 16x mode | — | — | 10 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 10 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_3B | | | Asynchronous SAMPR = 8x mode | — | — | 20 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 20 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_5B | | | Asynchronous SAMPR = 3x mode | — | — | 21 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 20 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_19B | | | Synchronous Mode X2 mode | — | — | 21 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 21 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_21B | | | Synchronous Mode X1 mode | — | — | 21 | Mbps | VDDIO = 3.3V, CLOAD = 30 pF(max) |
| | | | | — | — | 21 | Mbps | VDDIO = 1.8V, CLOAD = 30 pF(max) |
| UT_23 | FUSART | USART max GCLK_SERCOM | | — | — | FCLK_23 | MHz | — |
| UT_25 | FXCK | USART External Clock Input | | — | — | 25 | MHz | — |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.24. I²C Electrical Characteristics

**Figure 40-21.** I²C Start/Stop Bits Host Mode AC Timing Diagrams

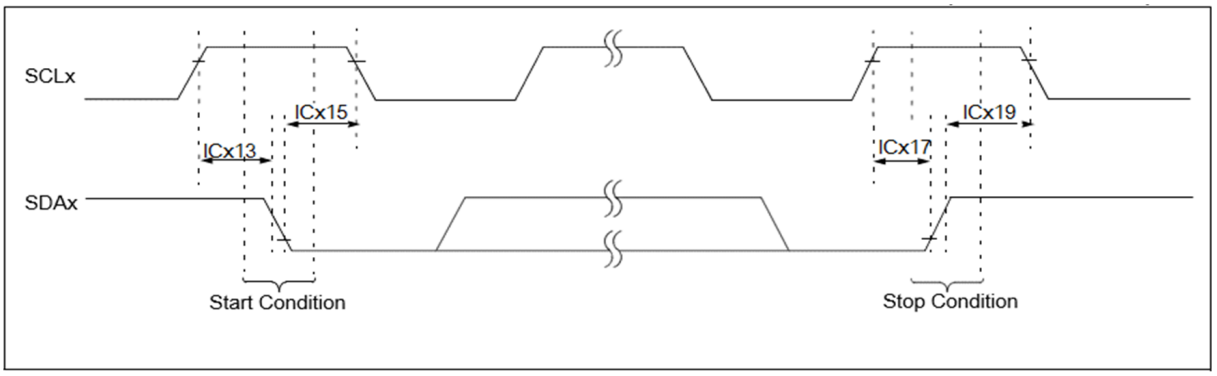



**Figure 40-22.** I²C Bus Data Host Mode AC Timing Diagrams

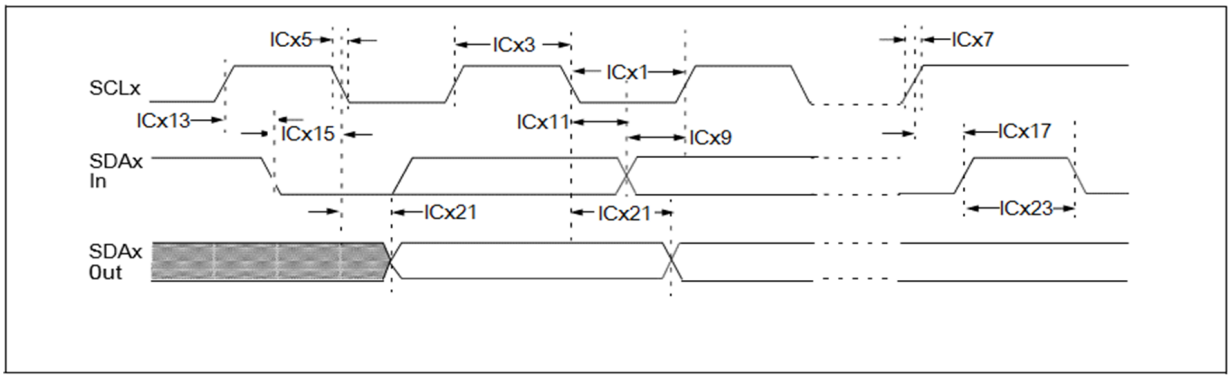



**Table 40-35.** I²C Host Mode AC Electrical Specifications

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics(1)** | | **Min.** | **Max.** | **Units** | **Conditions** |
| I2CM_1 | TL0:SCL | Host Clock Low Time | 100 kHz mode | 4.7 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | | | 1 MHz mode | 0.5 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-35.** I²C Host Mode AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics(1) | | Min. | Max. | Units | Conditions |
| I2CM_3 | THI:SCL | Host Clock High Time | 100 kHz mode | 4 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3 mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 60 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CM_5 | TF:SCL | SDAx and SCLx Fall Time | 100 kHz mode | — | 300 | ns | VDDIOx = 3.3V, IPULL-UP = 3 mA, CLOAD = 400 pF |
| | | | 400 kHz mode | — | 300 | ns | |
| | | | 1 MHz mode | — | 120 | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | — | 40 | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CM_7 | TR:SCL | SDAx and SCLx Rise Time | 100 kHz mode | — | 1000 | ns | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | — | 300 | ns | |
| | | | 1 MHz mode | — | 120 | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | — | 40 | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CM_9 | TSU:DAT | Data Setup Time | 100 kHz mode | 250 | — | ns | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 100 | — | ns | |
| | | | 1 MHz mode | 50 | — | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 10 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-35.** I²C Host Mode AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics(1) | | Min. | Max. | Units | Conditions |
| I2CM_11 | THD:DAT (1) | Data Hold Time | 100 kHz mode | 300 | — | ns | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 300 | — | ns | |
| | | | 1 MHz mode | 300 | — | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 5 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CM_13 | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4.7 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CM_15 | THD:STA | Start Condition Hold Time | 100 kHz mode | 4 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CM_17 | TSU:ST0 | Stop Condition Setup Time | 100 kHz mode | 4 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

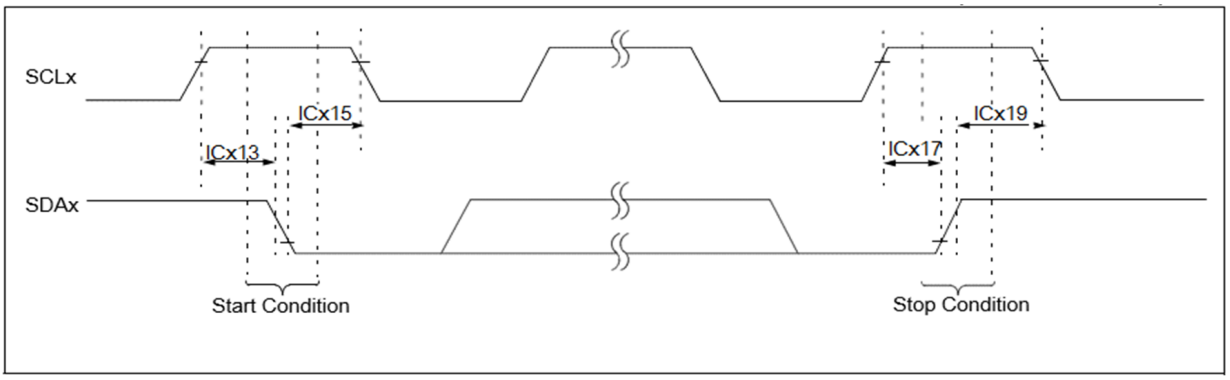
**Table 40-35.** I²C Host Mode AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics(1) | | Min. | Max. | Units | Conditions |
| I2CM_21 | TAA:SCL | Output Valid from Clock | 100 kHz mode | — | 3.45 | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | — | 0.9 | µs | |
| | | | 1 MHz mode | — | 0.45 | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | — | 100 | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CM_23 | TBF:SDA (2) | Bus Free Time | 100 kHz mode | 4.7 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | | | 1 MHz mode | 0.5 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |

**Notes:**

1. Longest delay between data hold timing based on bitfield SDAHOLD of register CTRLA from SERCOM Module and timing based on 4 period of GCLK_SERCOM for 100 kHz/400 kHz/1 MHz mode.
2. The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).

**Figure 40-23.** I²C Start/Stop Bits Client Mode AC Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-24.** I²C Bus Data Client Mode AC Timing Diagrams

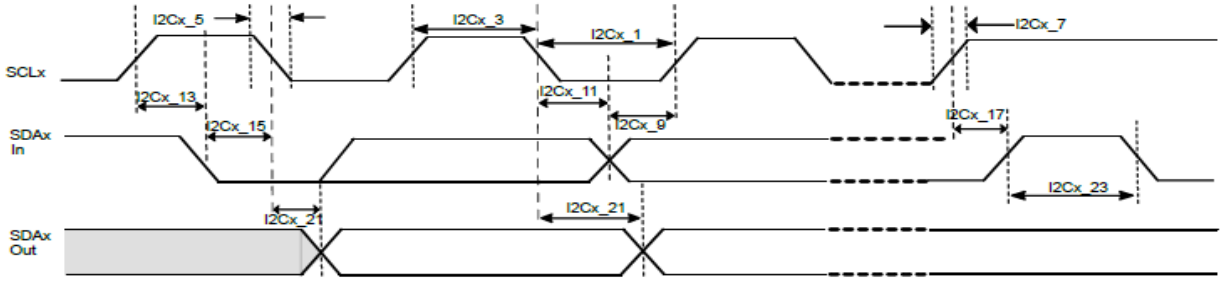



**Table 40-36.** I²C Client Mode AC Electrical Specifications

| AC CHARACTERISTICS_ | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics(1)** | | **Min.** | **Max.** | **Units** | **Conditions** |
| I2CS_1 | TL0:SCL | Client Clock Low Time | 100 kHz mode | 4.7 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | | | 1 MHz mode | 0.5 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_3 | THI:SCL | Client Clock High Time | 100 kHz mode | 4 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 60 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_5 | TF:SCL | SDAx and SCLx Fall Time | 100 kHz mode | — | 300 | ns | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | — | 300 | ns | |
| | | | 1 MHz mode | — | 120 | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | — | 40 | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-36.** I²C Client Mode AC Electrical Specifications (continued)

| AC CHARACTERISTICS_ | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics(1)** | | **Min.** | **Max.** | **Units** | **Conditions** |
| I2CS_7 | TR:SCL | SDAx and SCLx Rise Time | 100 kHz mode | — | 1000 | ns | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | — | 300 | ns | |
| | | | 1 MHz mode | — | 120 | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | — | 40 | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_9 | TSU:DAT | Data Setup Time | 100 kHz mode | 250 | — | ns | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 100 | — | ns | |
| | | | 1 MHz mode | 50 | — | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 10 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_11 | THD:DAT (1) | Data Hold Time | 100 kHz mode | 300 | — | ns | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 300 | — | ns | |
| | | | 1 MHz mode | 300 | — | ns | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 5 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_13 | TSU:STA | Start Condition Setup Time | 100 kHz mode | 4.7 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-36.** I²C Client Mode AC Electrical Specifications (continued)

| AC CHARACTERISTICS_ | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics(1)** | | **Min.** | **Max.** | **Units** | **Conditions** |
| I2CS_15 | THD:STA | Start Condition Hold Time | 100 kHz mode | 4 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_17 | TSU:ST0 | Stop Condition Setup Time | 100 kHz mode | 4 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 0.6 | — | µs | |
| | | | 1 MHz mode | 0.26 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_21 | TAA:SCL | Output Valid from Clock | 100 kHz mode | — | 3.45 | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | — | 0.9 | µs | |
| | | | 1 MHz mode | — | 0.45 | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | — | 100 | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |
| I2CS_23 | TBF:SDA (2) | Bus Free Time | 100 kHz mode | 4.7 | — | µs | VDDIOx = 3.3V, IPULL-UP = 3mA, CLOAD = 400 pF |
| | | | 400 kHz mode | 1.3 | — | µs | |
| | | | 1 MHz mode | 0.5 | — | µs | VDDIOx = 3.3V, IPULL-UP = 20 mA, CLOAD = 550 pF |
| | | | 3.4 MHz mode | 160 | — | ns | VDDIOx = 3.3V, IPULL-UP = 12 mA, CLOAD = 100 pF |

**Notes:**

1. Longest delay between data hold timing based on bitfield SDAHOLD of register CTRLA from SERCOM Module and timing based on 4 period of GCLK_SERCOM for 100 kHz/400 kHz/1 MHz mode.
2. The amount of time the bus must be free before a new transmission can start (STOP condition to START condition).


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.25. Controller Area Network (CAN) Electrical Characteristics

**Figure 40-25.** CANx Module AC Timing Diagram

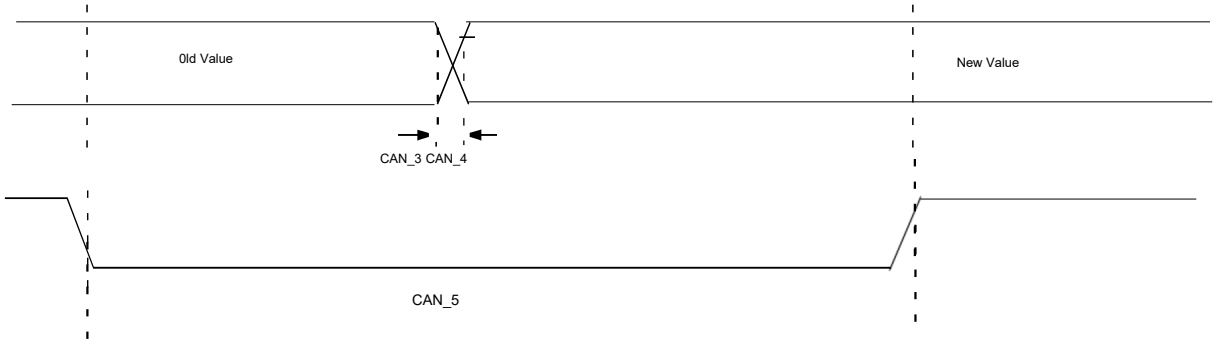


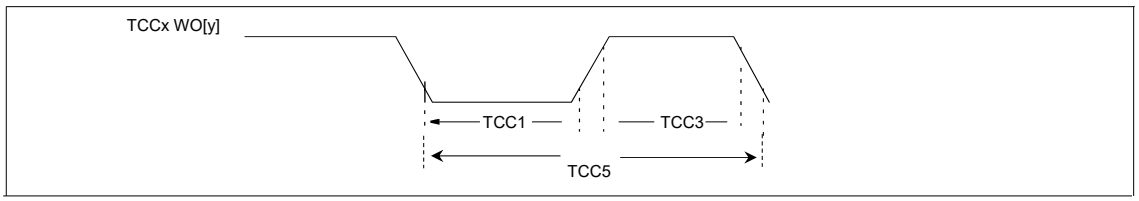

**Table 40-37.** CANx Module AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Conditions** |
| CAN_1 | CANRATE | CAN data rate | — | — | 10 | Mbps | VDDIOx = 3.3V,<br>CLOAD = 30 pF(max) |
| | | | — | — | 10 | | VDDIOx = 1.8V or VDDIO(min) whichever is greater,<br>CLOAD = 30 pF(max) |
| CAN_3 | CANFALL | Port Output Fall Time | — | — | DI_27 or DI_31 | ns | See parameter in I/O Specifications |
| CAN_4 | CANRISE | Port Output Rise Time | — | — | DI_25 or DI_29 | ns | See parameter in I/O Specifications |
| CAN_7 | fCAN_GCLK | CANx Input clk freq, GCLK_CAN | — | — | FCLK_25 | MHz | VDDIOx = 1.8V |

**Note:**
1. Assumes VDDIOx(min) and 30 pF external load on all CAN pins unless otherwise noted.

## 40.26. Timer/Counter Controller (TCC) Electrical Characteristics

**Figure 40-26.** TCCx Timer Capture Input Module AC Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 40-27.** TCCx Timer Compare Output Module AC Timing Diagrams

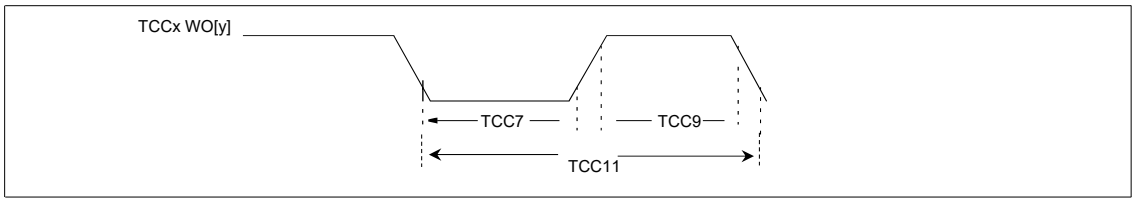



**Figure 40-28.** TCCx Timer Compare Fault Output Module AC Timing Diagrams

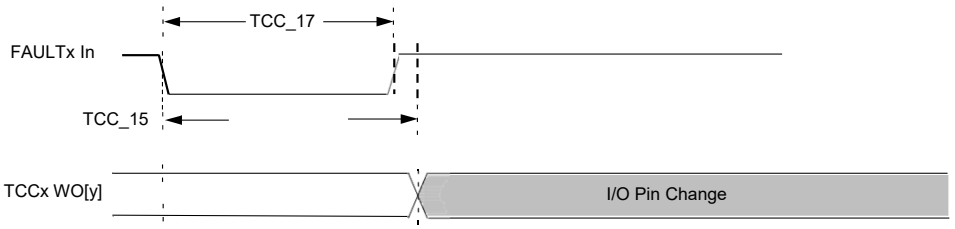



**Table 40-38.** TCCx Timer Capture Module AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| TCC_1 | TCCINLOW | Capture TCCx Input Low Time | 2/fGLK_TCCx | — | — | ns | VDDIOx(min) and meet TCC_5 spec |
| TCC_3 | TCCINHIGH | Capture TCCx Input High Time | 2/fGLK_TCCx | — | — | ns | VDDIOx(min) and meet TCC_5 spec |
| TCC_5 | TCCINPERIOD | Capture Input Period | 4/fGLK_TCCx | — | — | ns | VDDIOx(min) |
| TCC_7 | TCCOUTLOW | Compare TCCx Output Low Time | 3*DI_27 or 3*DI_31 | — | — | ns | VDDIOx(min) and meet TCC_11 spec |
| TCC_9 | TCCOUTHIGH | Compare TCCx Output High Time | 3*DI_25 or 3*DI_29 | — | — | ns | VDDIOx(min) and meet TCC_11 spec |
| TCC_11 | TCCOUTPERIOD | Compare Output Period | TCC_7+TCC_9 | — | — | ns | VDDIOx(min) |
| TCC_13 | fGCLK_TCCx | TCC peripheral module clock frequency | — | — | FCLK_35 | MHz | |
| TCC_15 | TCCFD | Fault Input to I/O Pin Change | — | — | 43 | ns | |
| TCC_17 | TCCFLT | Fault Input Pulse Width | 1/2*(t_rise+t_fall) | — | — | ns | |

**Note:**
1. Assumes VDDIOx(min) and 20 pF external load on TCC Fault Input I/O pin.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.27. Universal Serial Bus (USB) Electrical Characteristics

**Table 40-39.** USB AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO 3.0V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics[1] | Min. | Typ. | Max. | Units | Conditions |
| USB_1 | VDDUSB | USB Transceiver Voltage | 3 | — | 3.6 | V | Voltage on VDDIOx must be in this range for proper USB operation |
| **Clocks** | | | | | | | |
| USB_23 | USBCLKS [2] | USB Clock Source | — | 48 | — | MHz | fGCLK_USBFS = 48 MHz |
| USB_25 | USBAHB | Min. AHB Clock for USB Operations | 12 | — | — | MHz | |

**Notes:**

1. Supports Full and Low-Speed communication. Compliant with USB 2.0 specification.
2. Required clock accuracy: Full-Speed = ±0.25%, Low-Speed = ±1.5%.

**Table 40-40.** USB FS/LS Clock configuration

| USB FS/LS Clock configuration | | | Standard Operating Conditions: VDDIO 3.0V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | |
|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Device Mode Operation | Host Mode Operation |
| USB_30 | USBFS_GCLK_SRC | DFLL48M Open loop | No | No |
| | | DFLL48M Close loop, Ref. internal OSC source | No | No |
| | | DFLL48M Close loop, Ref. external XOSC / XOSC32K source | Yes | No |
| | | DFLL48M Close loop, Ref. SOF (USB recovery mode) | Yes | N/A |
| | | PLL internal OSC (DFLL48M) | No | No |
| | | PLL external OSC | Yes | Yes |
| | | XOSC Crystal | Yes | Yes |

## 40.28. Flash Controller Electrical Characteristics

**Table 40-41.** Flash NVM AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| NVM_1 | FRETEN | Flash Data Retention | 20 | — | — | Yrs | Under all conditions less than Absolute Maximum Ratings specifications |
| NVM_3 | EP [2] | Cell Endurance (Flash Erase and Write Operation) [2] | 10k | — | — | Cycles | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-41.** Flash NVM AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | | Min. | Typ. | Max. | Units | Conditions |
| NVM_5 | FREAD (1) | Flash Read | 0 Wait States | — | — | 24 | MHz | VDDIOx = 3.3V, FCR.CTRLA.AUTOWS = 0 |
| | | | 1 Wait States | — | — | 48 | | |
| | | | 2 Wait States | — | — | 72 | | |
| | | | 0 Wait States | — | — | 24 | MHz | VDDIOx = 1.8V or VDDIO(min) whichever is greater, FCR.CTRLA.AUTOWS = 0 |
| | | | 1 Wait States | — | — | 48 | | |
| | | | 2 Wait States | — | — | 72 | | |
| NVM_7 | TFW | Program Cycle Time | Write Double Word | — | — | 20 | µs | Under all conditions less than Absolute Maximum Ratings specifications |
| NVM_8 | TFPP | | Pre-Program Double Word | — | — | 3.5 | µs | |
| NVM_9 | TCE | | Erase Chip | — | — | 20 | ms | |
| NVM_11 | TFEP | | Erase Page | — | — | 20 | ms | |
| NVM_13 | IDDPROG | Supply Current during Programming | | — | — | PAI_401 | mA | VDDIOx = 3.3V |

**Notes:**

1. Maximum FLASH 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 FWS of the FCW CTRLA register when automatic wait states (AUTOWS) is disabled. This field defines the number of Wait states required to access the Embedded Flash Memory.
2. Cell Endurance is reached with Flash Preprogramming option disabled. Enabling such option will improve the Cell Endurance but slow-down the programming time.

## 40.29. Frequency Meter (FREQM) Electrical Characteristics

**Table 40-42.** Frequency Meter AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| FM_1 | FMLOW | GCLK_IOx Input Low Time | FMPERIOD/2 | — | — | ns | VDDIOx (min) and meet FM5 spec |
| FM_3 | FMHIGH | GCLK_IOx Input High Time | FMPERIOD/2 | — | — | ns | |
| FM_5 | FMPERIOD | GCLK_IOx Input Period | 1/FCLK_45 | — | — | ns | VDDIOx(min) |
| FM_7 | fGCLK_FREQM_REF | FREQM Reference | — | — | FCLK_17 | MHz | |
| FM_9 | fGCLK_FREQM_MSR | FREQM Measure | — | — | FCLK_15 | MHz | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.30. JTAG Electrical Characteristics

**Figure 40-29.** JTAG AC Timing Diagrams

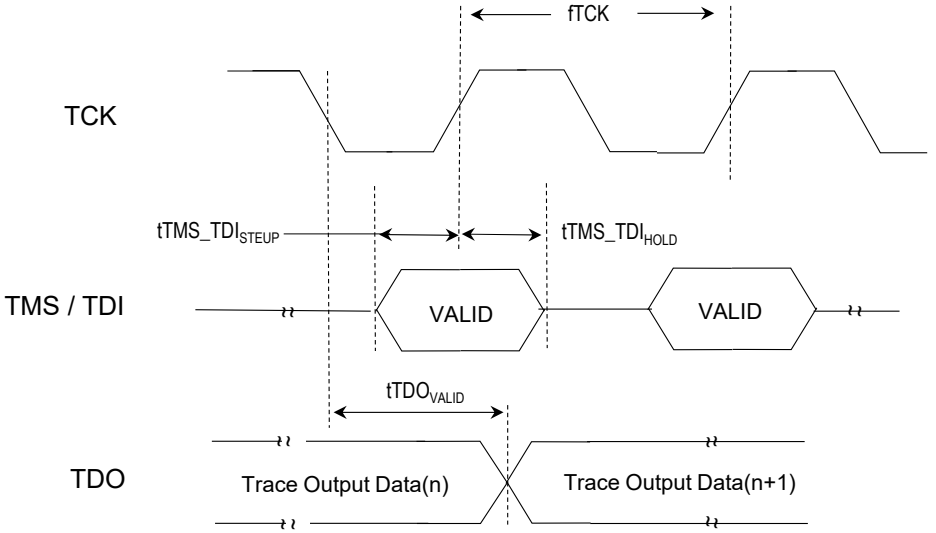



**Table 40-43.** JTAG AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| JTAG_1 | fTCK | JTAG TCK Frequency | — | — | 24 | MHz | VDDIO(min) - VDDIO(max) |
| JTAG_3 | tTMS_TDISETUP | TMS/TDI Setup Time | 2.3 | — | — | ns | |
| JTAG_5 | tTMS_TDIHOLD | TMS.TDI Hold Time | 0 | — | — | ns | |
| JTAG_7 | tTDOVALID | TDO Output Valid | — | — | 20.6 | ns | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 40.31. SWD 2-Wire Electrical Characteristics

**Figure 40-30.** SWD 2-Wire Read/Write AC Timing Diagrams

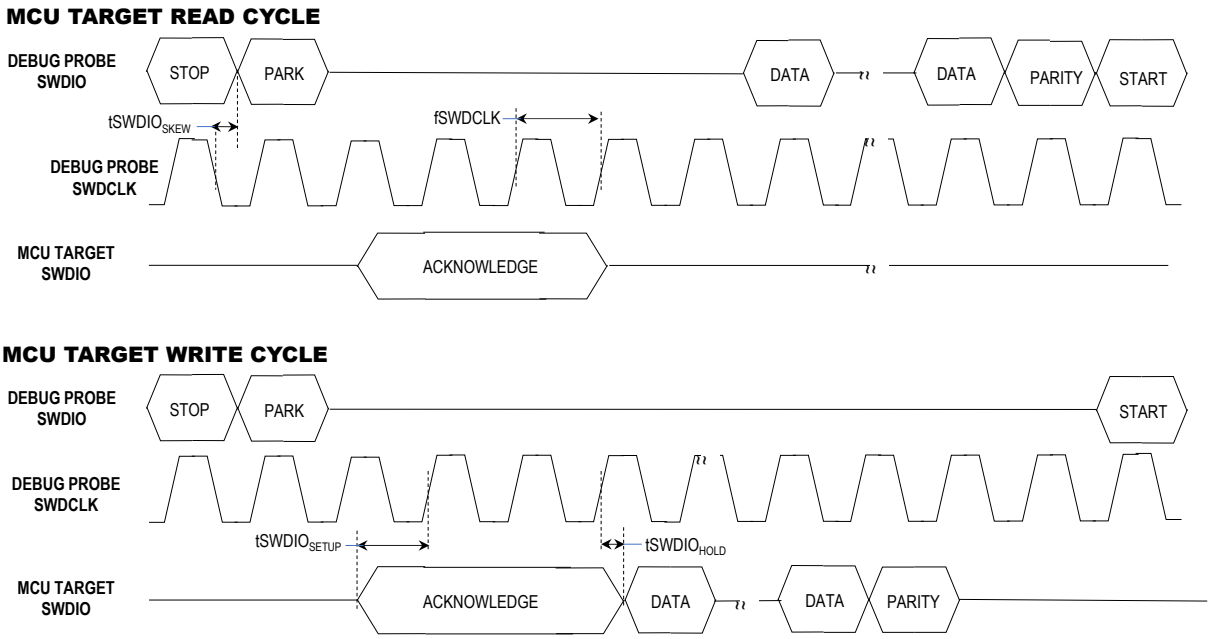



**Table 40-44.** SWD 2-Wire AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| SWD_1 | fSWDCLK | SWDCLK Clock Frequency | — | — | 40 | MHz | VDDIO(min) - VDDIO(max) |
| SWD_3 | tSWDCLKHIGH | SWDCLK Clock High Time | 1 / (2 * fSWDCLK) | — | — | ns | |
| SWD_5 | tSWDCLKLOW | SWDCLK Clock Low Time | 1 / (2 * fSWDCLK) | — | — | ns | |
| SWD_7 | tSWDIOSKEW | SWDIO Skew | -5 | — | 5 | ns | |
| SWD_9 | tSWDIOSETUP | SWDIO Setup Time | 4 | — | — | ns | |
| SWD_11 | tSWDIOHOLD | SWDIO Hold Time | 1 | — | — | ns | |

## 40.32. Configurable Custom Logic (CCL) Electrical Specifications

**Table 40-45.** CCL AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| CCL_1 | TSU_CCL_FF | Setup for CCL data to clock for flip-flop modes | 2*CCL_GCLK | — | — | ns | — |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 40-45.** CCL AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| CCL_3 | fGCLK_CCL | CCL peripheral module clock freq | — | — | FCLK_43 | MHz | — |

## 40.33. User LDO Electrical Specifications

**Table 40-46.** User LDO AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| ULDO_1 | VIN - VOUT | Dropout Voltage | — | — | 0.4 | V | IOUT = 100mA |
| ULDO_3 | | | — | — | 0.2 | V | IOUT = 30mA |
| ULDO_5 | VOUT | Output Voltage Accuracy | -5 | ±2 | 5 | % | IOUT = 100µA, ULDOLEVEL 0 (Vout 1.2V) |
| | | | -5 | ±2 | 5 | % | IOUT = 100µA, ULDOLEVEL 1 (Vout 1.5V) |
| | | | -5 | ±2 | 5 | % | IOUT = 100µA, ULDOLEVEL 2 (Vout 1.8V) |
| | | | -5 | ±2 | 5 | % | IOUT = 100µA, ULDOLEVEL 3 (Vout 2.5V) |
| ULDO_7 | ΔVOUT/VOUT | Load Regulation | — | 4.6 | 6.5 | % | 0mA ≤ IOUT ≤ 100 Ma, ULDOLEVEL 0 (Vout 1.2V) |
| | | | — | 3.7 | 5 | % | 0mA ≤ IOUT ≤ 100 mA, ULDOLEVEL 1 (Vout 1.5V) |
| | | | — | 3 | 4.1 | % | 0mA ≤ IOUT ≤ 100 mA, ULDOLEVEL 2 (Vout 1.8V) |
| | | | — | 2.2 | 3 | % | 0mA ≤ IOUT ≤ 100 mA, ULDOLEVEL 3 (Vout 2.5V) |
| ULDO_9 | IOUT | Output current | 0 | — | 100 | mA | — |
| ULDO_11 | ICL | Current Limit | 110 | 120 | 180 | mA | VOUT = 0V |

## 40.34. Multi-Channel Ram Controller (MCRAMC) Electrical Specifications

**Table 40-47.** MCRAMC Wait State Electrical Characteristics

| DC CHARACTERISTICS | | | Standard Operating Conditions: VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +85°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param No. | Symbol | Characteristics | Min. | Typ | Max. | Units | Conditions |
| MCRAMC_1 | FCLK3 (1) | MCRAMC maximum frequency | — | — | 0 to 72 | MHz | MCRAMC.CTRLA[WS] = 0<br>No RAM wait state introduced, maximum throughput operation |

**Note:**

1. Programmed via the MCRAMC.CTRLA[WS].


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 41. Electrical Characteristics 125°C

The specifications for Extended Temperature are identical to those shown in Electrical Characteristics, with the exception of the parameters listed in this chapter.

## 41.1. Absolute Maximum Ratings

**Table 41-1.** Absolute Maximum Electrical Characteristics

| | |
|---|---|
| Ambient temperature under bias | -40°C to +125°C |
| Storage temperature | -65°C to 150°C |
| Voltage on AVDD / VDDIO / VDDREG with respect to VSS | -0.3V to 4.03V |
| Voltage on any pin that is not 5V tolerant, with respect to VSS (3, 4) | -0.3V to +(VDDIO+0.3V) |
| Voltage on any 5V tolerant pin with respect to VSS when VDDIO ≥ 1.9V (3, 4) | -0.3V to 5.5V |
| Voltage on any 5V tolerant pin with respect to VSS when VDDIO < 1.9V (3, 4) | -0.3V to +(VDDIO+0.3V) |
| Voltage on USBFS D+ or D- pin with respect to VSS | -0.3V to +(VDDIO+0.3V) |
| Voltage on VREFH with respect to VSS | -0.3V to +(AVDD+0.3V) |
| Maximum current out of a VSS pins (2) | 200 mA |
| Maximum current into a VDDIO pins (2) | 200 mA |
| Maximum DC output current sourced/sunk by any 4x I/O pin (5, 6) | 15 mA |
| Maximum DC output current sourced/sunk by any 8x I/O pin (5, 6) | 25 mA |
| Maximum output current sunk/sourced by any port cluster (7) | 200 mA |
| Maximum Junction Temperature | 150°C |
| **ESD qualification:** | |
| Human Body Model (HBM) per JESD22-A114 | 4000V |
| Charged Device Model (CDM) (ANSI/ESD STM 5.3.1) | 1000V |

**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.
3. See the Table 40-17 for the 5V tolerant pins and 4x/8x pins.
4. When applying higher or lower voltage than those specified on I/O pins, refer to DI_19/DI_21 to respect the maximum injection current specification.
5. See the Table 46-17 for the I/O pin Drive Strength.
6. This maximum value is specific to continuous DC current. For I/O pin AC current characteristics, refer to the ‘I/O Pin Electrical Specifications’.
7. A cluster is a group of IO pins bounded by power/ground pins. Refer to the following table 'I/O Clusters'.

**Table 41-2.** I/O Clusters

| IO Cluster | |
|---|---|
| 1 | PD06, PD07, PD19, PD04, PD05, PD08, PD20, PD09, PA00, PA14, PA01, PA15, PA02 |
| 2 | PA10, PA11, PA12, PA16, PA17, PA18, PA03, PA04, PA05, PA06, PA13 |
| 3 | PA07, PA08, PA09, PA20, PA19, PB00, PB07, PB01, PB02 |
| 4 | PB11, PB12, PB13, PB14, PB03, PB04, PB05, PB08, PB09, PB06 |
| 5 | PB15, PB16, PB17, PB10, PC00, PC11, PC12, PC01, PC13 |
| 6 | PC02, PC14, PC15, PC03, PC16, PC09, PC04, PC17, PC10 |
| 7 | PC18, PC19, PC05, PC06, PC07, PC08, PD00, PC10, PC14, PD11, PD01 |
| 8 | PD02, PD03, PD15, PD16, PD17, PD12, PD18, PD13 |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 41.2. Operating Frequencies and Thermal Specifications

**Table 41-3.** Operating Frequency versus Voltage

| Param. No. | VDDIOx, AVDD Range | Temp. Range (in °C) | Max MCU Frequency | Comments |
|---|---|---|---|---|
| DC_7 | 1.71 to 3.63V | -40°C to +125°C | FCLK_1 MHz | Extended |

**Table 41-4.** MCU Thermal Operating Conditions

| Rating | Symbol | Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|
| Extended Temperature Range<br>Operating Ambient Temperature Range<br>Operating Junction Temperature Range | TA<br>TJ | -40<br>--- | — | 125<br>145 | °C<br>°C |
| Power Dissipation:<br>Internal Chip Power Dissipation:<br>PINT = (VDDIOx x (IDD –∑ IOHVDDIOx))+ (AVDD x (IDDANA - ∑ IOHAVDD)) + (VDDREG x IDDREG) I/O<br>Pin Power Dissipation:<br>PI/O = ∑ ((VDDIOx – VOHVDDIOx) x IOHVDDIOx) + ∑ (VOL x IOLVDDIOx) + ∑ ((AVDD – VOHAVDD) x IOHAVDD) + ∑ (VOL x IOLAVDD) | PD | PINT + PI/O | | | W |
| Maximum Allowed Power Dissipation | PDMAX | (TJ – TA)/θJA | | | W |

**Table 41-5.** Thermal Packaging Characteristics (1)

| Characteristics | Symbol | Typ. | Max. | Unit |
|---|---|---|---|---|
| Thermal Resistance, 48-pin TQFP (7x7x1 mm) Package | θJA | 47.0 | — | °C/W |
| Thermal Resistance, 64-pin TQFP (10x10x1 mm) Package | θJA | 44.8 | — | °C/W |
| Thermal Resistance, 100-pin TQFP (14x14x1 mm) Package | θJA | 41.1 | — | °C/W |
| Thermal Resistance, 48-pin VQFN (7x7x0.9 mm) Package | θJA | 22.3 | — | °C/W |
| Thermal Resistance, 64-pin VQFN (9x9x1 mm) Package | θJA | 20.5 | — | °C/W |

**Note:**
1. Junction to ambient thermal resistance, Theta-JA (θJA) numbers are achieved by package simulations.

## 41.3. MCU Active Power

**Table 41-6.** MCU Active Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | |
|---|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Clock/Freq** | **Power Pin** | **Typ. (1)** | **Max.** | **Units** | **Conditions** |
| APWR_1 | IDD_ACTIVE (2,3) | MCU IDD in active mode | PLL_48_Mhz | VDDIO = AVDD = 3.3V | 2.3 | 2.6 | mA | PLL runs at 48 MHz, uses XOSC with 48 MHz crystal as reference clock |
| APWR_3 | | | | VDDREG = 3.3V | 9.1 | 21.7 | mA | |
| APWR_5 | | | DFLL _48_MHz | VDDIO = AVDD = 3.3V | 0.8 | 2 | mA | DFLL in open loop |
| APWR_7 | | | | VDDREG = 3.3V | 4.9 | 17.3 | mA | |
| APWR_9 | | | XOSC _48_MHz | VDDIO = AVDD = 3.3V | 1.7 | 2.6 | mA | 48 MHz crystal |
| APWR_11 | | | | VDDREG = 3.3V | 5.4 | 18.4 | mA | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 41-6.** MCU Active Current Consumption DC Electrical Specifications (continued)

| DC CHARACTERISTICS | | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Clock/Freq | Power Pin | Typ. (1) | Max. | Units | Conditions |

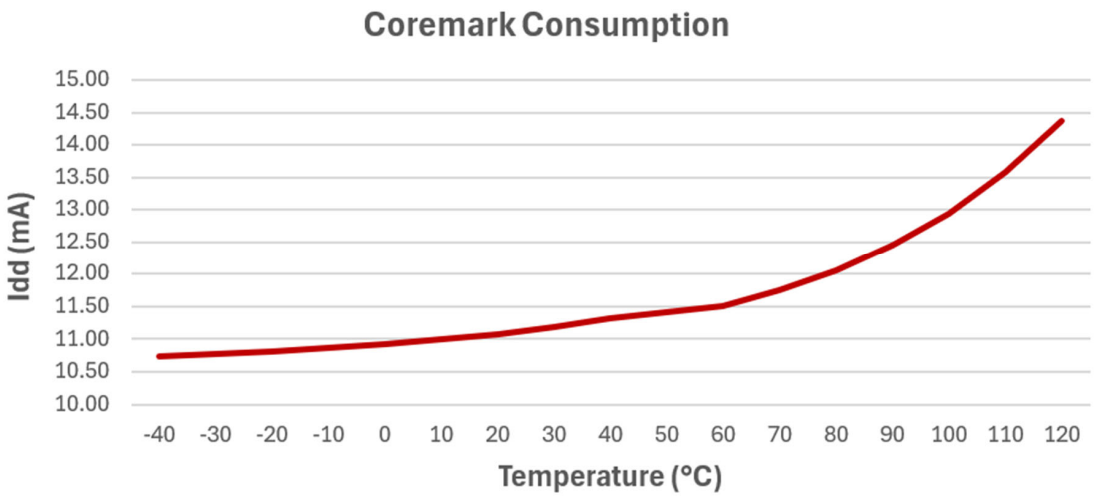
**Notes:**

1. Typical values at 25°C only.
2. Conditions:
   - RTC running on ULP32K for Coremark® ticks count
   - No other peripheral modules are operating (i.e. all peripherals except RTC inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled
3. MCU Running CoreMark® Test Suite.

Operating conditions:

- VDDIO = 3.3V
- PLL_48_MHz

**Figure 41-1.** Power Consumption over Temperature in Active Mode (Typical values for guidance only, not tested)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 41.4. Idle Power

**Table 41-7.** MCU Idle Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated) **Operating temperature:** **-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | |
|---|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Clock/Freq** | **Power Pin** | **Typ. (1)** | **Max.** | **Units** | **Conditions** |
| IPWR_1 | IDD_IDLE (2) | MCU IDD in IDLE mode | PLL_48_Mhz | VDDIO = AVDD = 3.3V | 1.4 | 1.6 | mA | PLL runs at 48 MHz, uses XOSC with 48 MHz crystal as reference clock |
| IPWR_3 | | | | VDDREG = 3.3V | 3.8 | 17.5 | mA | |
| IPWR_5 | | | DFLL _48_MHz | VDDIO = AVDD = 3.3V | 0.8 | 1 | mA | DFLL in open loop |
| IPWR_7 | | | | VDDREG = 3.3V | 1.8 | 15.8 | mA | |
| IPWR_9 | | | XOSC _48_MHz | VDDIO = AVDD = 3.3V | 1.4 | 1.5 | mA | 48 MHz crystal |
| IPWR_11 | | | | VDDREG = 3.3V | 2.1 | 16.1 | mA | |

**Notes:**

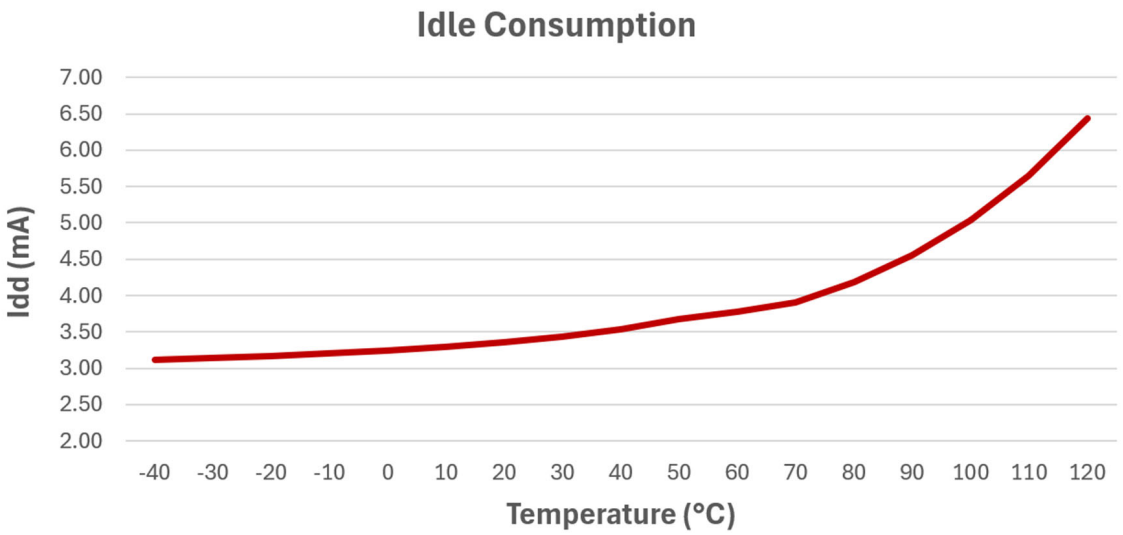
1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating (i.e. all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

Operating conditions:

- VDDIO = 3.3V
- PLL_48_Mhz


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-2.** Power Consumption over Temperature in IDLE Mode (Typical values for guidance only, not characterized over Process / voltage)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 41.5. Standby Power

**Table 41-8.** MCU Standby Power Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Power Pin | Typ. (1) | Max. | Units | Conditions |
| SPWR_5 | IDD_STANDBY (2) | MCU IDD in STANDBY mode w/VDDCORE_x = 0.8V | VDDIO = AVDD = 3.3V | 0.22 | 0.37 | mA | RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1) Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0) |
| SPWR_7 | | | VDDREG = 3.3V | 0.47 | 7.7 | mA | |
| SPWR_9 | | | VDDIO = AVDD = 3.3V | 0.13 | 0.26 | mA | RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVSTDBY = 0x1) Voltage Reference Low Voltage enabled (SUPC.VREF.LPSTDBY = 0x1) |
| SPWR_11 | | | VDDREG = 3.3V | 0.16 | 7.4 | mA | |
| SPWR_17 | | MCU IDD in STANDBY mode w/VDDCORE_x = 1.2V | VDDIO = AVDD = 3.3V | 0.25 | 0.41 | mA | RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0) Voltage Reference Low Voltage disabled (SUPC.VREF.LPSTDBY = 0x0) |
| SPWR_19 | | | VDDREG = 3.3V | 0.7 | 18 | mA | |
| SPWR_21 | | | VDDIO = AVDD = 3.3V | 0.16 | 0.3 | mA | RAM Low Power mode enabled (PM.STDBYCFG.LPRAM = 0x1), NVM Power Reduction Mode is Enter Auto Standby (FCR.CTRLB.SLP = 0x0), Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVSTDBY = 0x0) Voltage Reference Low Voltage enabled (SUPC.VREF.LPSTDBY = 0x1) |
| SPWR_23 | | | VDDREG = 3.3V | 0.33 | 18.1 | mA | |

**Notes:**

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating, unless specified (i.e. all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

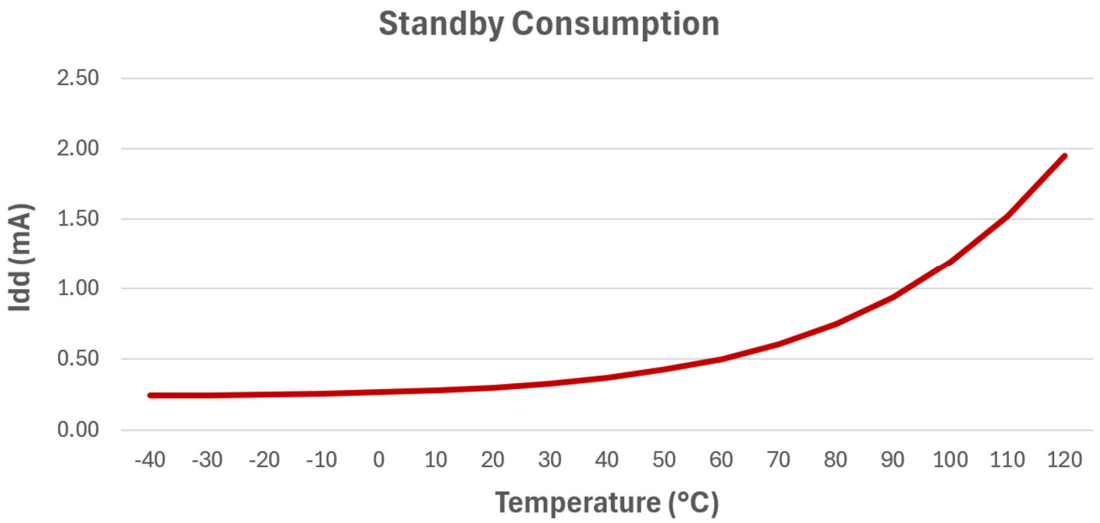
Operating Conditions:

- VDDIO = 3.3V
- RAM Low Power mode enabled


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- NVM Power Reduction Mode is Enter Auto Standby
- Voltage Regulators Low Voltage enabled
- Voltage Reference Low Voltage enabled
- No RTC

**Figure 41-3.** Power Consumption over Temperature in Standby Mode (Typical values for guidance only, not tested)





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 41.6. Hibernate Power

**Table 41-9.** MCU Hibernate Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Power Pin | Typ. (1) | Max. | Units | Conditions |
| HPWR_8 | IDD_HIBERNATE (2) | MCU IDD in HIBERNATE mode w/ VDDCORE_x = 0.8V | VDDIO = AVDD = 3.3V | 101 | 156 | µA | RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1),<br>Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)<br>Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0)<br>RTC running on XOSC32K |
| HPWR_9 | | | VDDREG = 3.3V | 200 | 2420 | µA | |
| HPWR_12 | | | VDDIO = AVDD = 3.3V | 19 | 57 | µA | RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1),<br>Voltage Regulators Low Voltage enabled (SUPC.VREGCTRL.LVHIB = 0x1)<br>Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1)<br>RTC running on XOSC32K |
| HPWR_13 | | | VDDREG = 3.3V | 48 | 2310 | µA | |
| HPWR_18 | | MCU IDD in HIBERNATE mode w/ VDDCORE_x = 1.2V | VDDIO = AVDD = 3.3V | 101 | 158 | µA | RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1),<br>Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0)<br>Voltage Reference Low Voltage disabled (SUPC.VREFCTRL.LPHIB = 0x0)<br>RTC running on XOSC32K |
| HPWR_19 | | | VDDREG = 3.3V | 292 | 5490 | µA | |
| HPWR_22 | | | VDDIO = AVDD = 3.3V | 19 | 57 | µA | RAM Low Power mode enabled (PM.HIBCFG.LPRAM = 0x1),<br>Voltage Regulators Low Voltage disabled (SUPC.VREGCTRL.LVHIB = 0x0)<br>Voltage Reference Low Voltage enabled (SUPC.VREFCTRL.LPHIB = 0x1)<br>RTC running on XOSC32K |
| HPWR_23 | | | VDDREG = 3.3V | 101 | 5410 | µA | |

**Notes:**

1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating, unless specified (i.e. all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

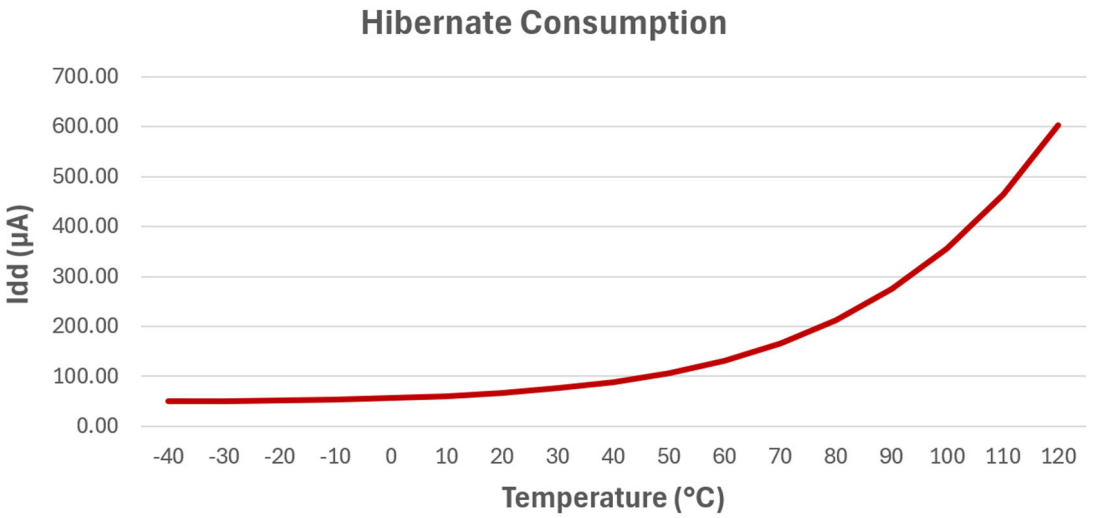
Operating conditions:

- VDDIO = 3.3V
- Voltage Regulators Low Voltage enabled
- Voltage Reference Low Voltage enabled
- RAM Low Power mode enabled


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

- No RTC

**Figure 41-4.** Power Consumption over Temperature in Hibernate Mode (Typical values for guidance only, not tested)




## 41.7. Backup Power

**Table 41-10.** MCU Backup Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Power Pin** | **Typ (1)** | **Max.** | **Units** | **Conditions** |
| BPWR_1 | IDD_BACKUP (2) | MCU IDD in BACKUP mode powered from VDDIOx | VDDIO = AVDD = 3.3V | 6.3 | 183 | µA | RTC running on XOSC32K (32.768KHz), TRAM enabled |
| BPWR_3 | | | VDDREG = 3.3V | 0.6 | 2.1 | µA | |

**Notes:**

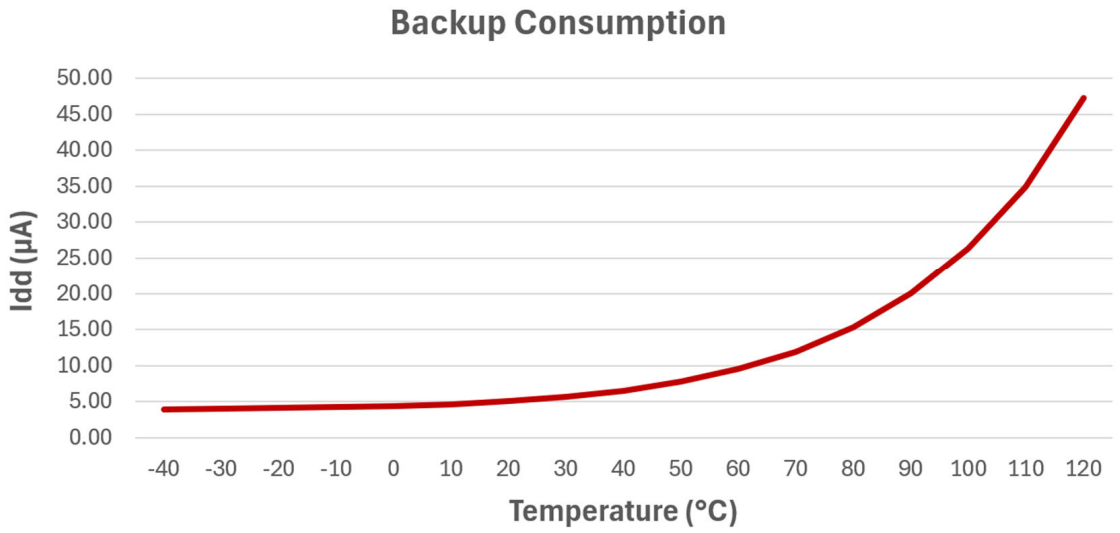
1. Typical values at 25°C only.
2. Conditions:
   - No peripheral modules are operating (i.e. all peripherals inactive)
   - APB Peripheral bus clocks: default settings after reset
   - MCU is running on Flash with automatic wait state
   - I/Os are inactive input mode with input trigger disabled
   - All clock generation sources disabled unless otherwise specified
   - WDT, CFD Clock Fail Detect disabled

Operating conditions:

- VDDIO = 3.3V
- No RTC


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-5.** Power Consumption over Temperature in Backup Mode (Typical values for guidance only, not tested)




## 41.8. OFF Power

**Table 41-11.** MCU OFF Current Consumption DC Electrical Specifications

| DC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Power Pin | Typ. (1) | Max. | Units | Conditions |
| OPWR_1 | IDD_OFF (2) | MCU IDD in OFF | VDDIO = VDDANA = 3.3V | 1 | 14 | µA | In Off mode, the device is entirely powered-off. (SLEEPCFG.SLEEPMODE = OFF), and subsequent execution of the WFI instruction.<br>**Note:** This mode is left by pulling the RESET pin low, or when a power Reset is done. |
| OPWR_3 | | | VDDREG = 3.3V | 0.03 | 1.52 | µA | |

**Notes:**

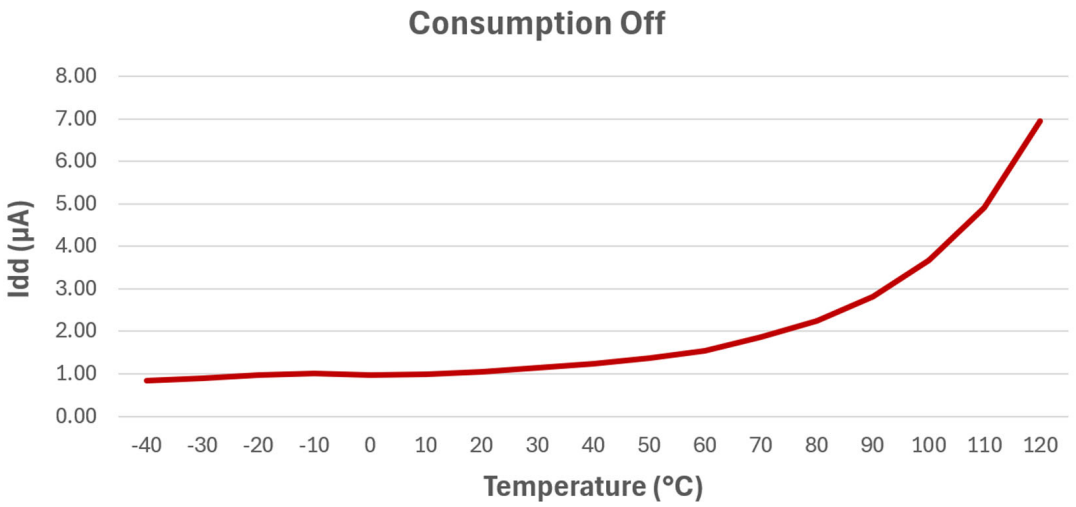
1. Typical values at 25°C only.
2. Conditions:
   - All peripherals inactive unless otherwise specified
   - All clock generation sources disabled unless otherwise specified
   - All I/O pins configured as input pins pulled down or tied to GND

Operating conditions:

- VDDIO = 3.3V


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-6.** Power Consumption over Temperature in Off Mode (Typical values for guidance only, not tested)




## 41.9. Maximum Clock Frequencies

**Table 41-12.** Maximum Clock Frequencies AC Electrical Specifications

| AC CHARACTERISTICS | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | |
|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Max** | **Units** |
| FCLK_1 | FCY | MCU clock freq | 48 | MHz |
| FCLK_3 | fAHB | AHB clock freq | 48 | MHz |
| FCLK_5 | fAPB0:1] | APB clock freq | 48 | MHz |
| FCLK_5 | fAPB2 | | 48 | MHz |
| FCLK_6 | fGCLKGEN[0] | GCLK clock freq output | 160 | MHz |
| | fGCLKGEN[1:7] | | 160 | MHz |
| | fGCLKGEN[8:11] | | 80 | MHz |
| FCLK_13 | fGCLK_EIC | EIC input clock freq | 80 | MHz |
| FCLK_15 | fGCLK_FREQM_MSR | FREQM Measure Clock Frequency | 160 | MHz |
| FCLK_17 | fGCLK_FREQM_REF | FREQM Reference Clock Frequency | 80 | MHz |
| FCLK_19 | fGCLK_EVSYS_CHANNELx | EVSYS channel x input clock freq | 80 | MHz |
| FCLK_23 | fGCLK_SERCOM_CORE | SERCOM input clock freq | 100 | MHz |
| FCLK_25 | fGCLK_CANx | CAN input clock freq | 80 | MHz |
| FCLK_35 | fGCLK_TCCx | TCCx input clock freq | 160 | MHz |
| FCLK_49 | fGCLK_AC | Analog comparator peripheral module clock freq | 80 | MHz |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 41.10. Analog Comparator (AC) Electrical Specifications

**Table 41-13.** Comparator AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| CMP_1 | VIOFF | Input Offset Voltage | -10 | — | 10 | mV | COMPCTRLn.HYST = 0x3.<br>Comparator ref voltage = AVDD/2 |
| CMP_5 | VHYST_00 | Input Hysteresis Voltage | 2 | — | 18 | mV | COMPCTRLn.HYST = 0x0.<br>Comparator ref voltage = AVDD/2 |
| | VHYST_01 | Input Hysteresis Voltage | 5 | — | 28 | mV | COMPCTRLn.HYST = 0x1.<br>Comparator ref voltage = AVDD/2 |
| | VHYST_02 | Input Hysteresis Voltage | 18 | — | 63 | mV | COMPCTRLn.HYST = 0x2.<br>Comparator ref voltage = AVDD/2 |
| | VHYST_03 | Input Hysteresis Voltage | 30 | — | 95 | mV | COMPCTRLn.HYST = 0x3.<br>Comparator ref voltage = AVDD/2 |
| CMP_15 | TRESPSS ain<0:1> | Small Signal Response Time - High speed | — | — | 133 | ns | COMPCTRLn.SPEED = 0x0<br>COMPCTRLn.HYST = 0x3<br>Comparator ref voltage = AVDD/2.<br>Input overdrive = +/- 180mV |
| | | Small Signal Response Time - Low Speed | — | — | 467 | ns | COMPCTRLn.SPEED = 0x1<br>COMPCTRLn.HYST = 0x3<br>Comparator ref voltage = AVDD/2.<br>Input overdrive = +/- 180mV |
| CMP_23 | CVREFRNG | Comparator Voltage Reference Input Range | | — | | V | (1) |
| CMP_25 | FGCLK_AC | Analog comparator peripheral module clock freq | — | — | FCLK_49 | MHz | See FCLK_49 in Maximum Clock Frequencies |

**Note:**

1. Comparator Ref voltage cannot exceed:
   VIN(max) - VIOFF(max) - CMP_5(max) - 100mV) ≥CMP_23 ≥ (VIN(min) + abs(VIOFF(min)) + CMP_5(max) + 100mV).

**Table 41-14.** DAC Module AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.6V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ.(1) | Max. | Units | Conditions |
| DAC_7 | VOUT | Output Voltage Range | AVSS | — | AVDD - 1LSB | V | — |
| DAC_17 | Tset_FS | DAC Full Scale Settling Time | — | — | 1.8 | µs | +/-4 LSB of final value for step size from 10% to 90% |

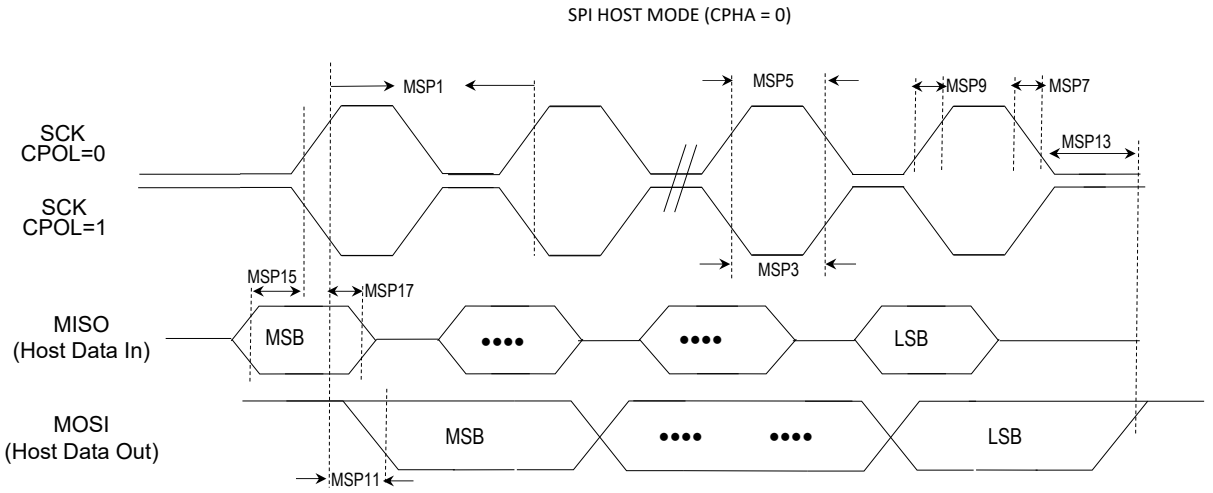

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 41-14.** DAC Module AC Electrical Specifications (continued)

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.6V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Industrial** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ.(1) | Max. | Units | Conditions |
| DAC_18a | Thold | DAC Holding Time | 0.25 | — | — | ms | This parameter applies only for Low-Power mode:<br>DACCTRLn.SHENm = 0x1<br>Time for 1 LSB voltage loss of the DAC output |
| DAC_18b | | | 0.31 | — | — | ms | This parameter applies only for Low-Power mode:<br>DACCTRLn.SHENm = 0x1<br>Time for 10 LSB voltage loss of the DAC output |
| SDAC_19 | INL | Integral Non Linearity | -0.5 | — | 0.5 | LSB | — |
| SDAC_21 | DNL | Differential Non Linearity | -0.4 | — | 0.4 | LSB | — |
| SDAC_23 | GERR | Gain Error | -2 | — | -0.2 | LSB | — |
| SDAC_25 | EOFF | Offset Error | -1.4 | — | 1.4 | LSB | Normal mode:<br>DACCTRLn.SHENm = 0x0 |
| | | | -6.4 | — | 6.4 | LSB | Low-Power mode:<br>DACCTRLn.SHENm = 0x1 |

## 41.11. Serial Peripheral Interface (SPI) Timings and Electrical Characteristics

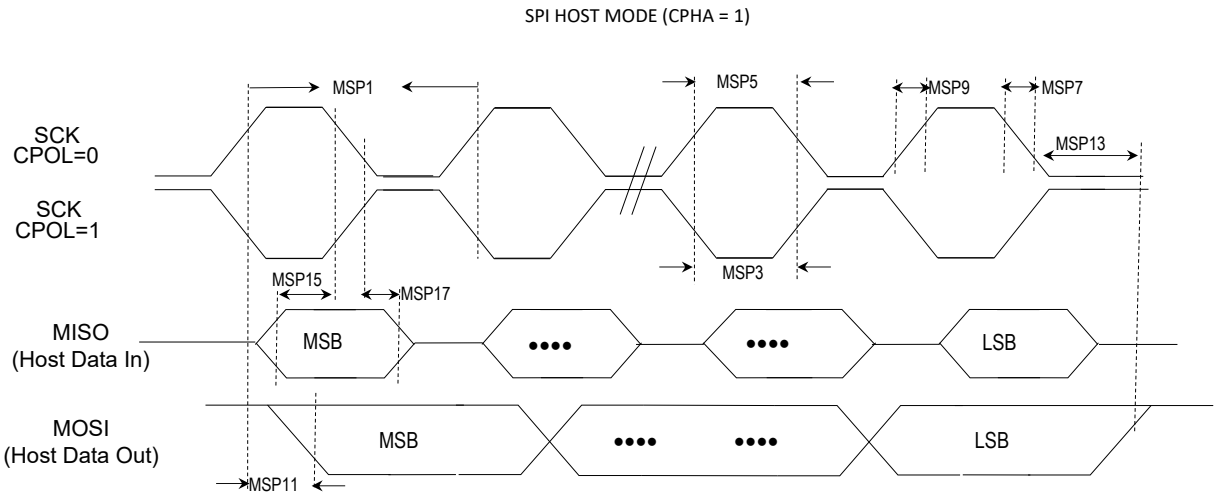
**Figure 41-7.** SPIx Host Module (CPHA = 0) Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-8.** SPIx Host Module (CPHA = 1) Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 41-15.** SPIx Module Host Mode Electrical Specifications [1]

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics[1] | | Min. | Typ. | Max. | Units | Conditions |
| MSP_1 | FSCK | SCK Frequency | Normal Speed SERCOM | — | — | 21 | MHz | Transmitter mode, CTRB.RXEN = 0.<br>CLOAD = 30 pF(MAX) |
| | | | | — | — | 21 | | Full Duplex Transmit and Receive mode, Loop Back mode with one SPI talking to another SPI on the same MCU.<br>CLOAD = 30 pF(MAX) |
| | | | | — | — | 1/(2*(TMIS+NOTE2_TV)) [2] | | Full Duplex Transmit and Receive mode,<br>CLOAD=30 pF (MAX).<br>The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the loop back mode above may therefore be possible using the formula. |
| | | | High Speed SERCOM [3] | — | — | 45 | | Transmitter mode, CTRB.RXEN = 0,<br>Dut talking to test chip in the same board VDDIOx >= 2.97V,<br>CLOAD = 20 pF(MAX) |
| | | | | — | — | 1/(2*(TMIS_HS+NOTE2_TV)) (2) | | Full Duplex Transmit and Receive mode,<br>CLOAD=20pF(MAX).<br>The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the loop back mode above may therefore be possible using the formula.<br>VDDIOx >= 2.97V. |
| MSP_11 | TMOV | MOSI Data Output Valid after SCK | Normal Speed SERCOM | | — | 9.56 | ns | — |
| | | | High Speed SERCOM [3] | | — | 2.95 | ns | VDDIOx >= 2.97V |
| MSP_13 | TMOH | MOSI hold after SCK | | 0 | — | — | ns | — |
| MSP_15A | TMIS | MISO Setup Time of Data Input to SCK - Normal Speed SERCOM | Normal Speed SERCOM | 13.24 | — | — | ns | — |
| | | | High Speed SERCOM [3] | 1.6 | — | — | ns | VDDIOx >= 2.97V |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

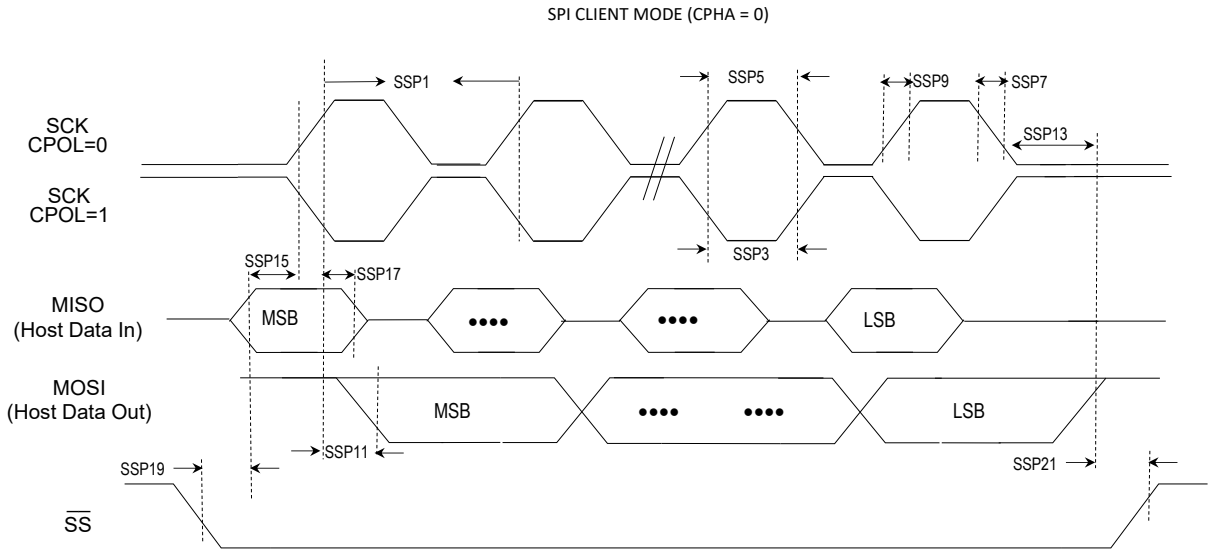
**Table 41-15.** SPIx Module Host Mode Electrical Specifications [1] (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG = VDDIO = AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics[1] | | Min. | Typ. | Max. | Units | Conditions |
| MSP_17 | TMIH | MISO Hold Time of Data Input to SCK | Normal Speed SERCOM | 0 | — | — | ns | — |
| | | | High Speed SERCOM [3] | 1.21 | — | — | ns | VDDIOx >= 2.97V |

**Notes:**

1. Assumes 30pF external load on all SPIx pins unless otherwise noted.
2. NOTE2_TV is the client external device data output valid time from clock edge specification.
3. SERCOM0 pins PC0, PC1, PC2, and PC3.

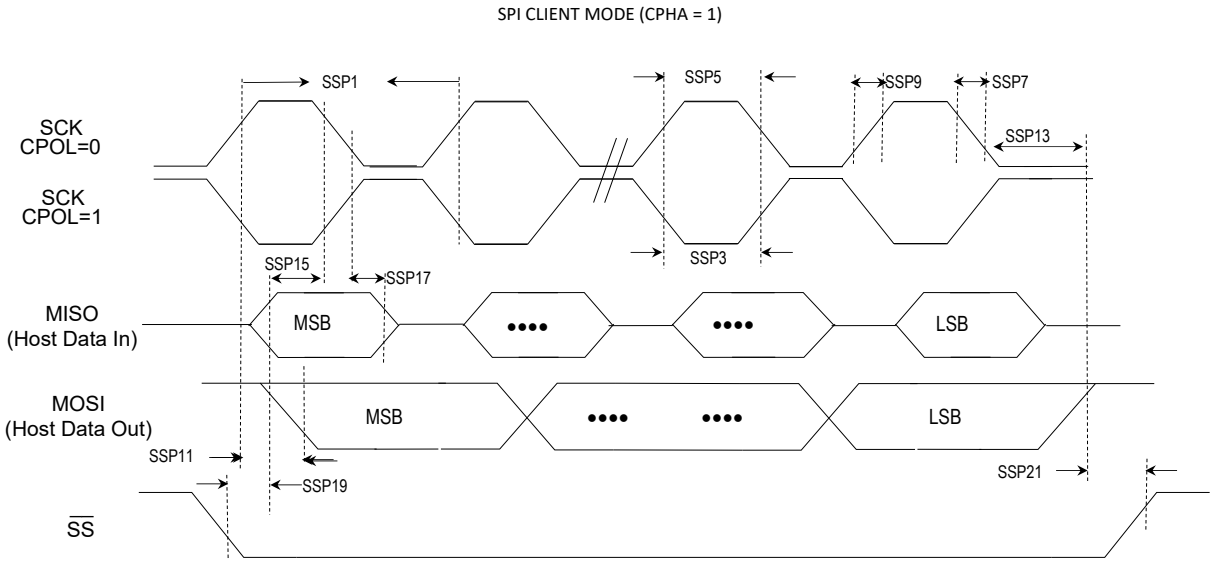
**Figure 41-9.** SPIx Client Module (CPHA = 0) Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-10.** SPIx Client Module (CPHA = 1) Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 41-16.** SPIx Module Client Mode Electrical Specifications

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics (1) | | Min. | Typical | Max. | Units | Conditions |
| SSP_1A | FSCK | SCK Frequency | Normal Speed SERCOM | — | — | 21 | MHz | Receiver mode.<br>CLOAD = 30 pF(MAX) |
| | | | | — | — | 21 | | Full Duplex Transmit and Receive mode,<br>Loop Back mode with one SPI talking to another SPI on the same MCU.<br>CLOAD = 30 pF(MAX) |
| | | | | — | — | 1/(2*(TSOV+NOTE2_TMIS)) (2) | | Full Duplex Transmit and Receive mode,<br>CLOAD=30pf(MAX).<br>The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the loop back mode above may therefore be possible using the formula. |
| | | | High Speed SERCOM (3) | — | — | 45 | | Receiver mode. DUT talking to test chip in the same board<br>VDDIOx>=2.97V,<br>CLOAD = 20 pF(MAX) |
| | | | | — | — | 1/(2*(TSOV_HS+NOTE2_TMIS)) (2) | | Full Duplex Transmit and Receive mode,<br>CLOAD=20pF(MAX).<br>The max SPI speed of the MCU is partially dependent on the external SPI device performance characteristics.<br>Faster speeds than the loop back mode above may therefore be possible using the formula.<br>VDDIOx>=2.97V |
| SSP_11A | TSOV | MISO Data Output Valid after SCK | Normal Speed SERCOM | — | — | 23.15 | ns | — |
| | | | High Speed SERCOM (3) | — | — | 10.81 | ns | VDDIOx>=2.97V |
| SSP_13 | TSOH | MISO hold after SCK | | 4 | — | — | ns | — |
| SSP_15 | TSIS | MOSI Setup Time of Data Input to SCK | Normal Speed SERCOM | 7.33 | — | — | ns | — |
| | | | High Speed SERCOM (3) | 2.23 | — | — | ns | VDDIOx>=2.97V |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 41-16.** SPIx Module Client Mode Electrical Specifications (continued)

| AC CHARACTERISTICS | | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | | |
|---|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics (1) | | Min. | Typical | Max. | Units | Conditions |
| SSP_17 | TSIH | MOSI Hold Time of Data Input to SCK | Normal Speed SERCOM | 2.28 | — | — | ns | — |
| | | | High Speed SERCOM (3) | 1.92 | — | — | ns | VDDIOx>=2.97V |
| SSP_19 | TSSS | $\overline{SS}$ setup to SCK (PRELOADEN=0) | Normal Speed SERCOM | 5.76 | — | — | ns | — |
| | | | High Speed SERCOM (3) | 1.5 | — | — | ns | VDDIOx>=2.97V |
| SSP_21 | TSSH | $\overline{SS}$ hold after SCK Client | Normal Speed SERCOM | 2.01 | — | — | ns | — |
| | | | High Speed SERCOM (3) | 2.04 | — | — | ns | VDDIOx>=2.97V |

**Notes:**

1. Assumes 30pF external load on all SPIx pins unless otherwise noted.
2. NOTE2_TMIS is the host external device setup time.
3. SERCOM0 pins PC0, PC1, PC2, and PC3.

## 41.12. Timer/Counter Controller (TCC) Electrical Characteristics

**Figure 41-11.** TCCx Timer Capture Input Module AC Timing Diagrams

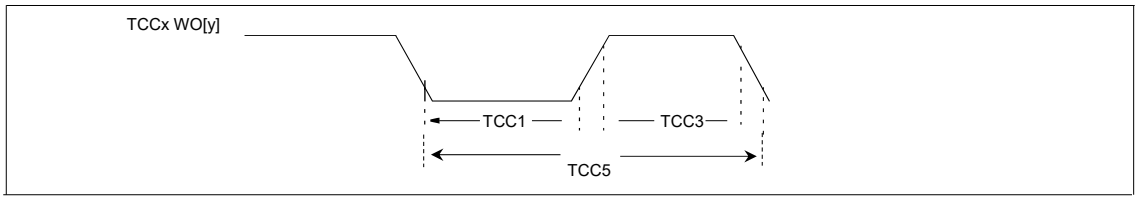
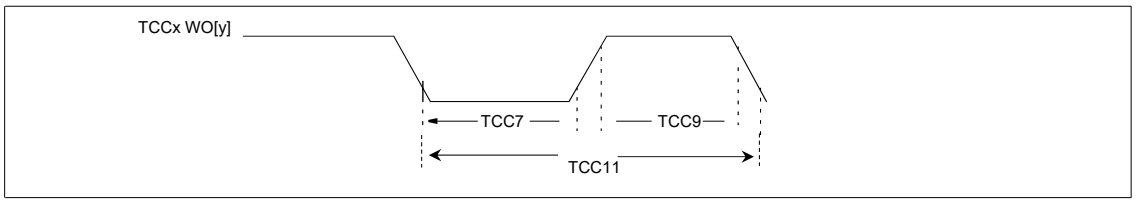



**Figure 41-12.** TCCx Timer Compare Output Module AC Timing Diagrams





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Figure 41-13.** TCCx Timer Compare Fault Output Module AC Timing Diagrams

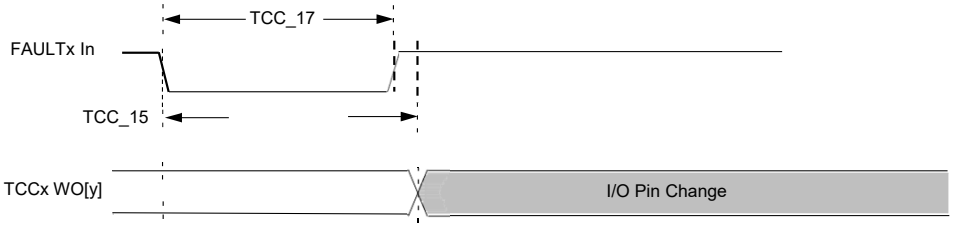



**Table 41-17.** TCCx Timer Capture Module AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | | |
|---|---|---|---|---|---|---|---|
| **Param. No.** | **Symbol** | **Characteristics** | **Min.** | **Typ.** | **Max.** | **Units** | **Conditions** |
| TCC_1 | TCCINLOW | Capture TCCx Input Low Time | 2/fGLK_TCCx | — | — | ns | VDDIOx(min) and meet TCC_5 spec |
| TCC_3 | TCCINHIGH | Capture TCCx Input High Time | 2/fGLK_TCCx | — | — | ns | VDDIOx(min) and meet TCC_5 spec |
| TCC_5 | TCCINPERIOD | Capture Input Period | 4/fGLK_TCCx | — | — | ns | VDDIOx(min) |
| TCC_7 | TCCOUTLOW | Compare TCCx Output Low Time | 3*DI_27 or 3*DI_31 | — | — | ns | VDDIOx(min) and meet TCC_11 spec |
| TCC_9 | TCCOUTHIGH | Compare TCCx Output High Time | 3*DI_25 or 3*DI_29 | — | — | ns | VDDIOx(min) and meet TCC_11 spec |
| TCC_11 | TCCOUTPERIOD | Compare Output Period | TCC_7+TCC_9 | — | — | ns | VDDIOx(min) |
| TCC_13 | fGCLK_TCCx | TCC peripheral module clock frequency | — | — | FCLK_35 | MHz | |
| TCC_15 | TCCFD | Fault Input to I/O Pin Change (1) | — | — | 43 | ns | |
| TCC_17 | TCCFLT | Fault Input Pulse Width | 1/2*(t_rise+t_fall) | — | — | ns | |

**Note:**
1. Assumes VDDIOx(min) and 20 pF external load on TCC Fault Input I/O pin.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 41.13. JTAG Electrical Specifications

**Figure 41-14.** JTAG AC Timing Diagrams

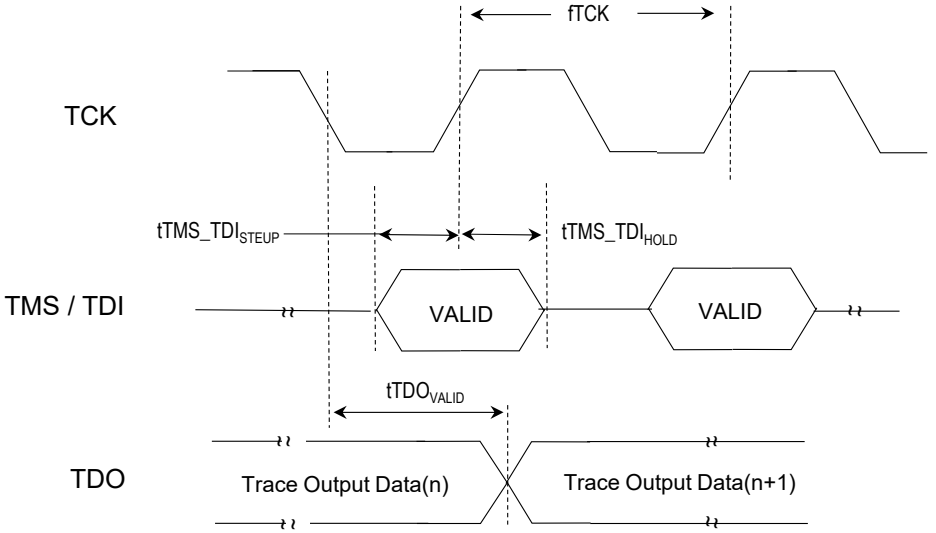



**Table 41-18.** JTAG AC Electrical Specifications

| AC CHARACTERISTICS | | | Standard Operating Conditions: VDDREG=VDDIO=AVDD 1.71V to 3.63V (unless otherwise stated)<br>**Operating temperature:**<br>**-40°C ≤ TA ≤ +125°C for Extended Temperature** | | | | |
|---|---|---|---|---|---|---|---|
| Param. No. | Symbol | Characteristics | Min. | Typ. | Max. | Units | Conditions |
| JTAG_1 | fTCK | JTAG TCK Frequency | — | — | 23 | MHz | VDDIO(min) - VDDIO(max) |
| JTAG_7 | tTDOVALID | TDO Output Valid | — | — | 21.13 | ns | |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 42. Packaging Information

## 42.1. Package Marking Information

All devices are marked with the Microchip logo, a shortened ordering code, the Arm logo, and additional information that will be marked as follows:

YYWWNNN

Where:

- "YY": Manufacturing Year
- "WW": Manufacturing Week
- "NNN": Internal Code

## 42.2. Package Drawings

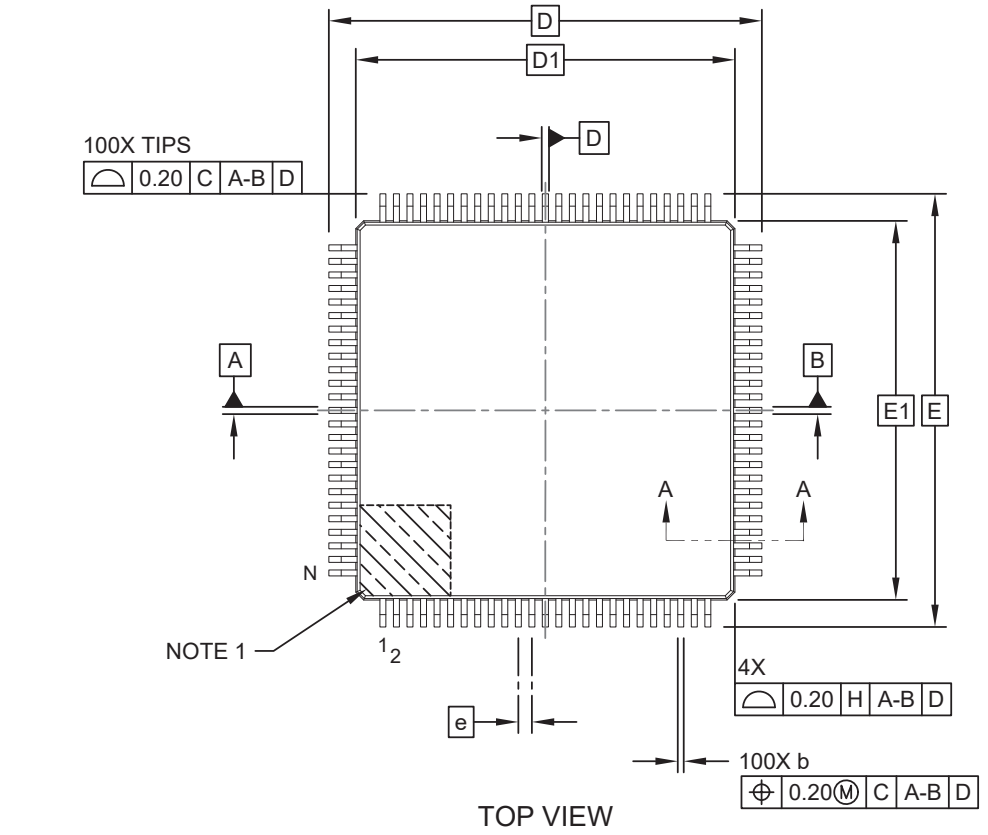
**Note:** For current package drawings, refer to the Microchip Packaging Specification, which is available at www.microchip.com/packaging.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

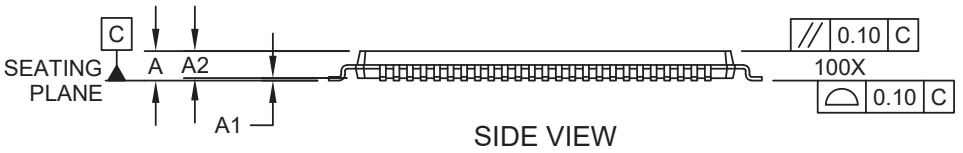
### 42.2.1. 100-Pin TQFP

**100-Lead Plastic Thin Quad Flatpack (E5X) - 14x14x1 mm Body [TQFP]; With 2.00 mm Footprint**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|




TOP VIEW



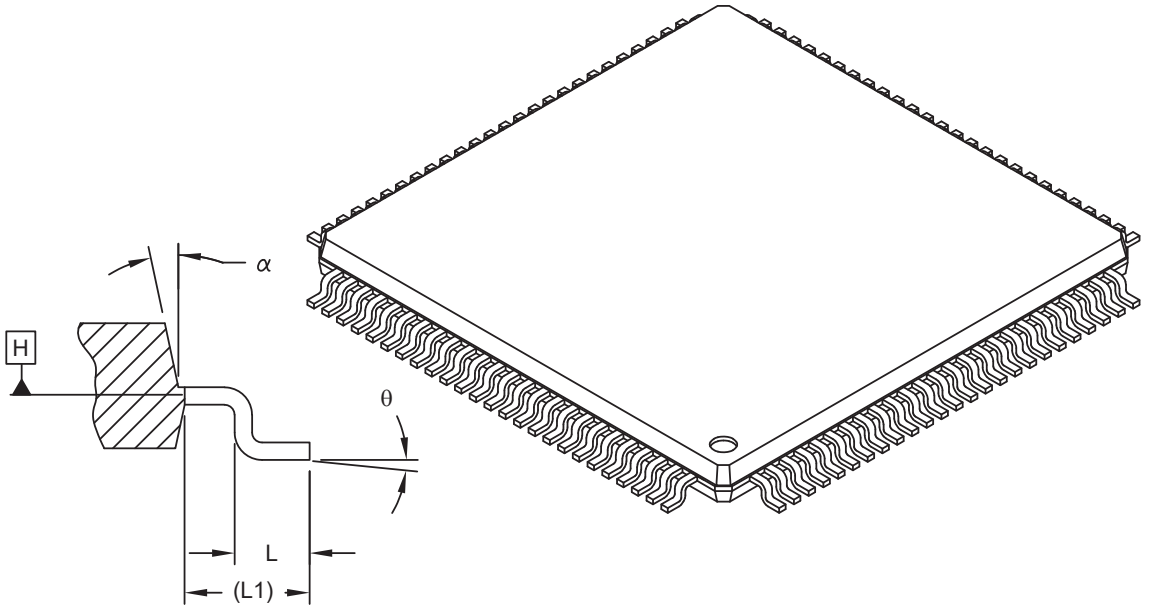

SIDE VIEW

Microchip Technology Drawing C04-00110-E5X Rev D Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 100-Lead Plastic Thin Quad Flatpack (E5X) - 14x14x1 mm Body [TQFP]; With 2.00 mm Footprint

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




SECTION A-A

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Leads | N | | 100 | |
| Lead Pitch | e | | 0.50 BSC | |
| Overall Height | A | - | - | 1.20 |
| Standoff | A1 | 0.05 | - | 0.15 |
| Molded Package Thickness | A2 | 0.95 | 1.00 | 1.05 |
| Foot Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | L1 | | 1.00 REF | |
| Foot Angle | θ | 0° | - | 7° |
| Overall Width | E | | 16.00 BSC | |
| Overall Length | D | | 16.00 BSC | |
| Molded Package Width | E1 | | 14.00 BSC | |
| Molded Package Length | D1 | | 14.00 BSC | |
| Lead Width | b | 0.17 | 0.22 | 0.27 |
| Mold Draft Angle Top | α | 11° | - | 13° |

Notes:

1. The Pin 1 visual index feature may vary, but it must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M.

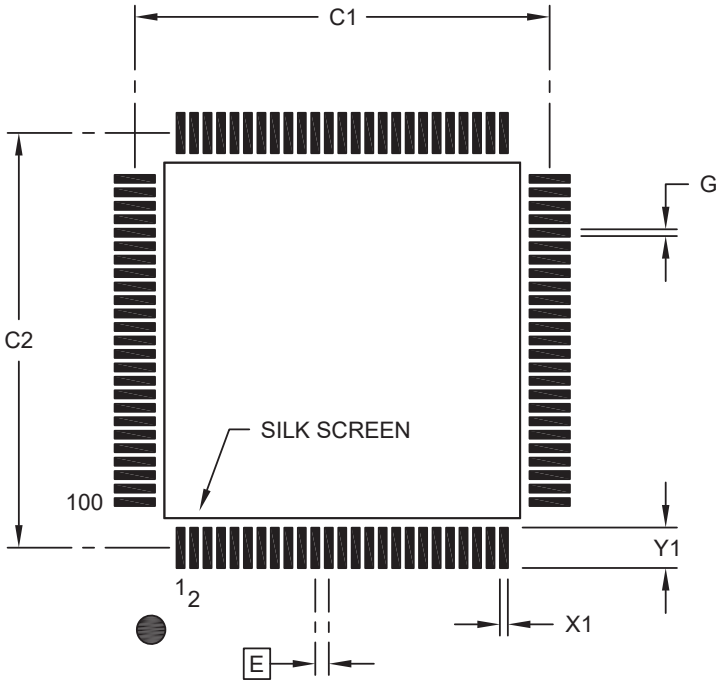
BSC: Basic Dimension. The theoretically exact value is shown without tolerances.
REF: Reference Dimension, usually without tolerance, is for information purposes only.




© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 100-Lead Plastic Thin Quad Flatpack (E5X) - 14x14x1 mm Body [TQFP]; With 2.00 mm Footprint

**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 | | 15.40 | |
| Contact Pad Spacing | C2 | | 15.40 | |
| Contact Pad Width (X100) | X1 | | | 0.30 |
| Contact Pad Length (X100) | Y1 | | | 1.50 |
| Contact Pad to Contact Pad (X96) | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. The theoretically exact value is shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during the reflow process.

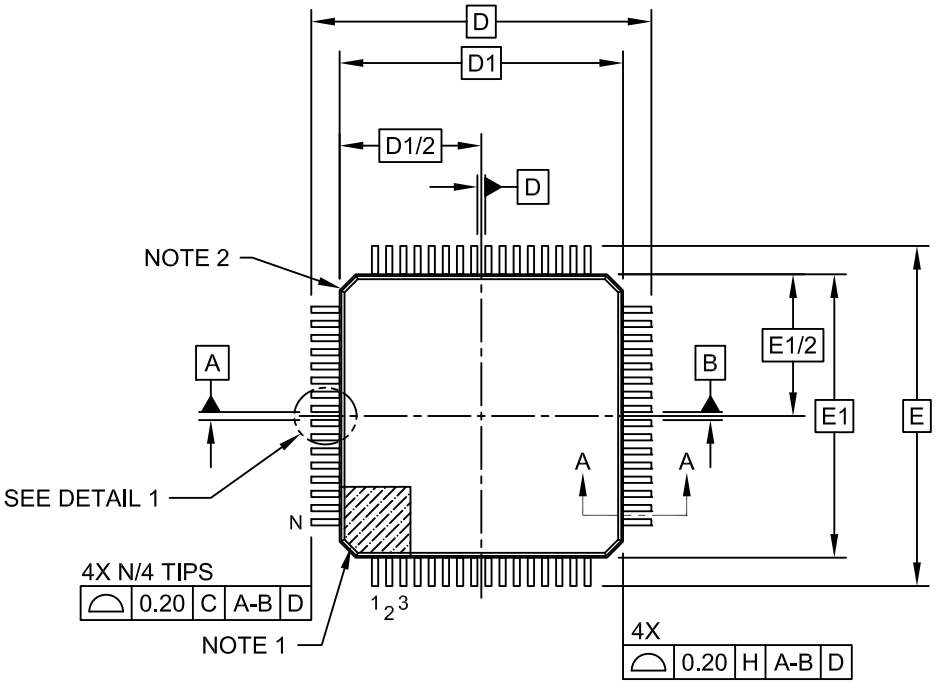
Microchip Technology Drawing C04-02110-E5X Rev D


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

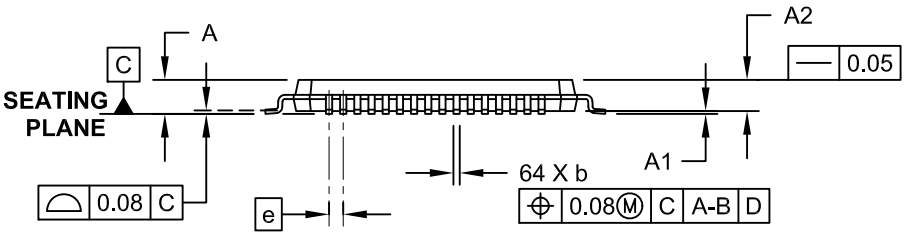
### 42.2.2. 64-Pin TQFP

**64-Lead Plastic Thin Quad Flatpack (V2X)-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



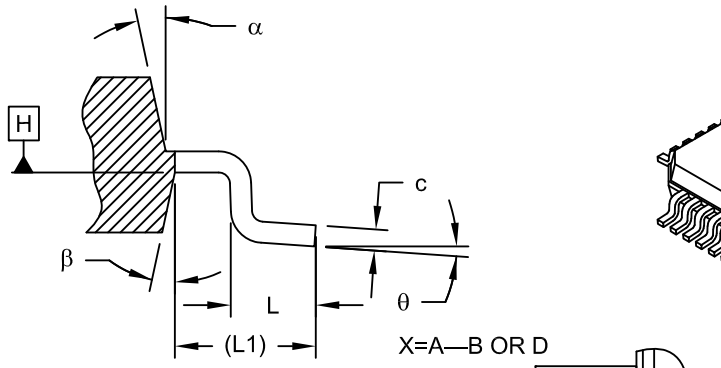

SIDE VIEW

Microchip Technology Drawing C04-085-V2X Rev E Sheet 1 of 2

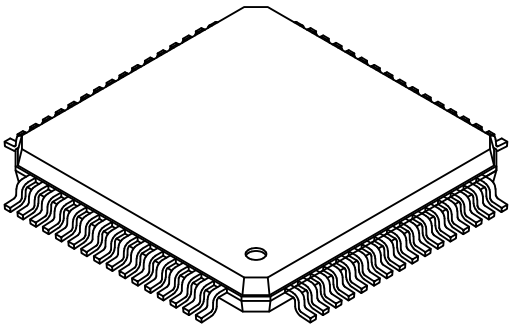

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

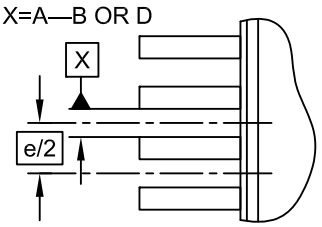
## 64-Lead Plastic Thin Quad Flatpack (V2X)-10x10x1 mm Body, 2.00 mm Footprint [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




SECTION A-A






DETAIL 1

| 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 | θ | 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:

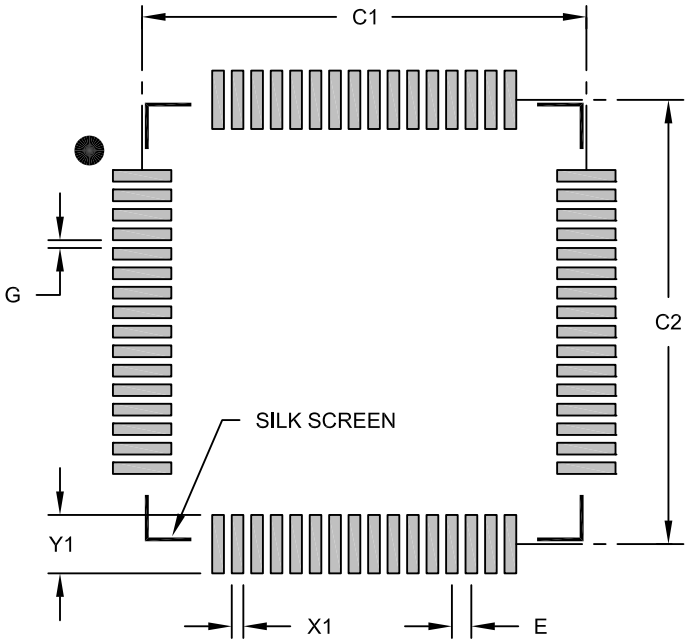
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085-V2X Rev E Sheet 2 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Plastic Thin Quad Flatpack (V2X) 10x10x1 mm Body, 2.00 mm Footprint [TQFP]




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 (X64) | X1 | | | 0.30 |
| Contact Pad Length (X64) | 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.

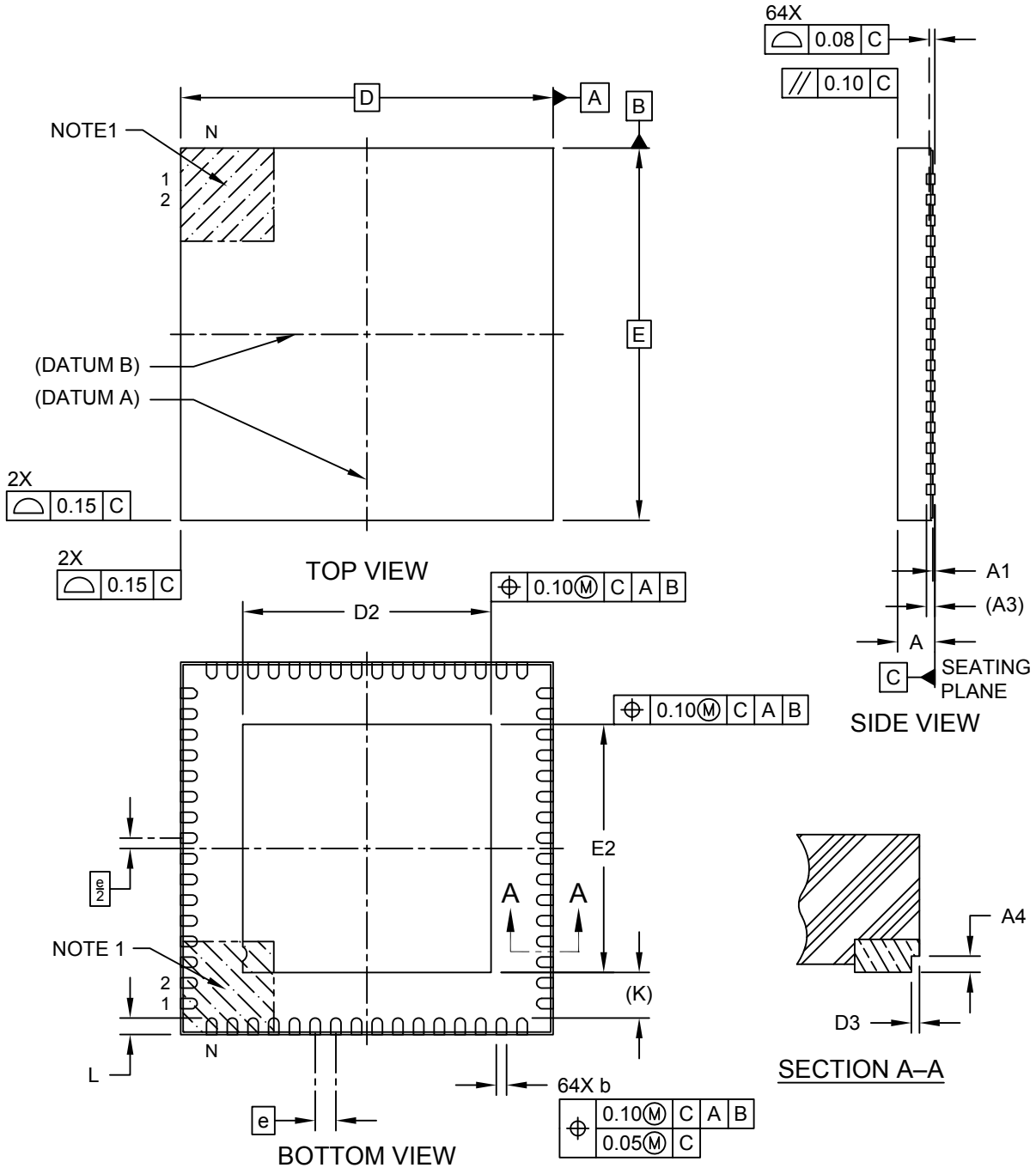
Microchip Technology Erawing C04-2085-V2X Rev E


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.2.3. 64-Pin VQFN

**64-Lead Very Thin Quad Flat, No Lead Package (LZX) - 9x9x0.9 mm Body [VQFN]**
**With 6.00 mm Exposed Pad and Stepped Wettable Flanks**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|

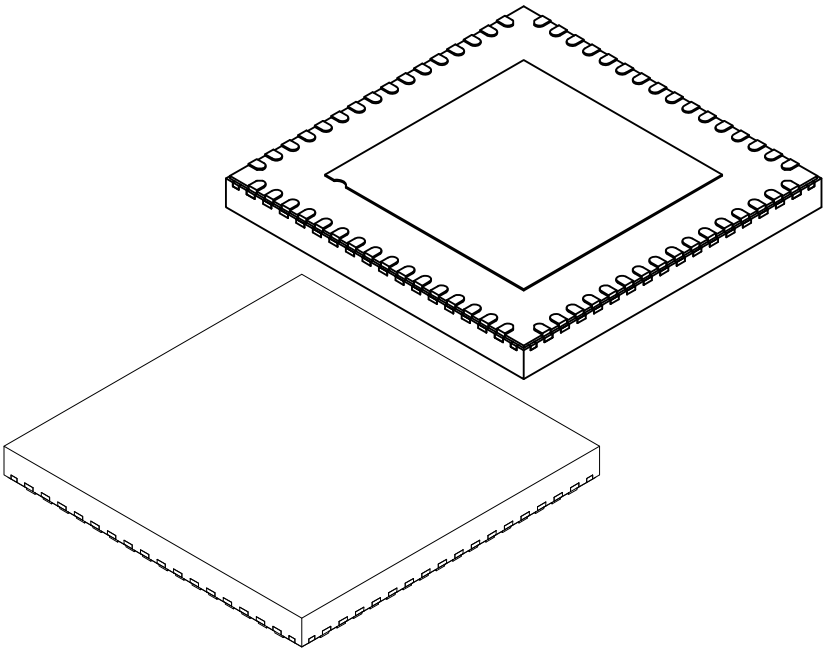



Microchip Technology Drawing C04-514 Rev A Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Very Thin Quad Flat, No Lead Package (LZX) - 9x9x0.9 mm Body [VQFN] With 6.00 mm Exposed Pad and Stepped Wettable Flanks

**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 | | 64 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | – | – | 0.90 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Terminal Thickness | A3 | | 0.203 REF | |
| Overall Length | D | | 9.00 BSC | |
| Exposed Pad Length | D2 | 5.90 | 6.00 | 6.10 |
| Overall Width | E | | 9.00 BSC | |
| Exposed Pad Width | E2 | 5.90 | 6.00 | 6.10 |
| Terminal Width | b | 0.18 | 0.25 | 0.30 |
| Terminal Length | L | 0.30 | 0.40 | 0.50 |
| Terminal-to-Exposed-Pad | K | | 1.10 REF | |
| Wettable Flank Step Length | D3 | - | - | 0.085 |
| Wettable Flank Step Height | A4 | 0.10 | - | 0.19 |

Notes:

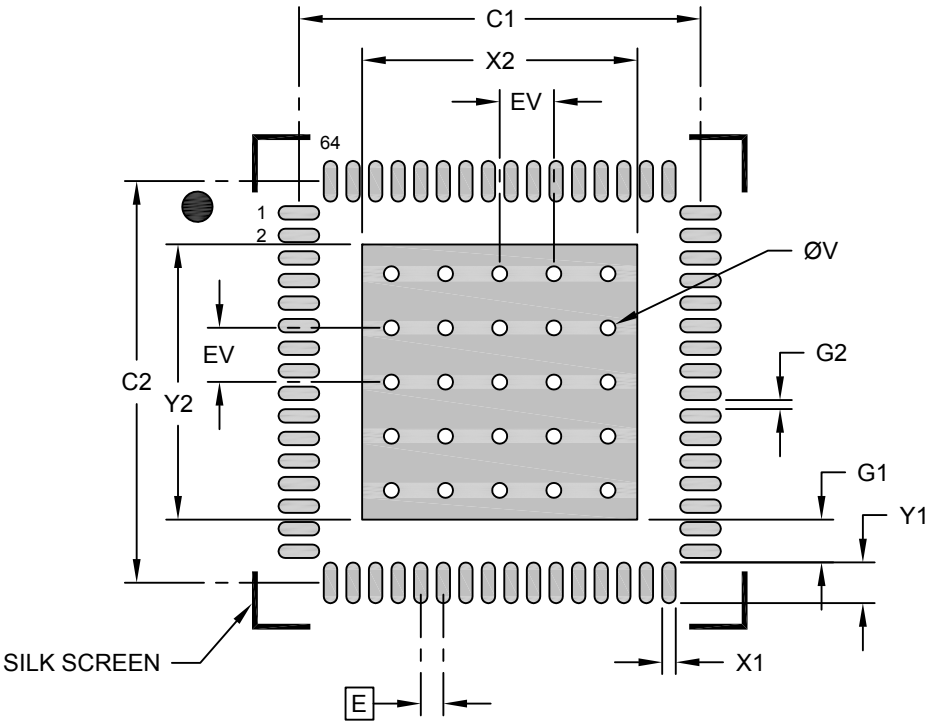
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-514 Rev A Sheet 2 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Very Thin Quad Flat, No Lead Package (LZX) - 9x9x0.9 mm Body [VQFN] With 6.00 mm Exposed Pad and Stepped Wettable Flanks

**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 | | | 6.10 |
| Optional Center Pad Length | Y2 | | | 6.10 |
| Contact Pad Spacing | C1 | | 8.90 | |
| Contact Pad Spacing | C2 | | 8.90 | |
| Contact Pad Width (Xnn) | X1 | | | 0.30 |
| Contact Pad Length (Xnn) | Y1 | | | 0.90 |
| Contact Pad to Center Pad (Xnn) | G1 | 0.95 | | |
| Contact Pad to Contact Pad (Xnn) | 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

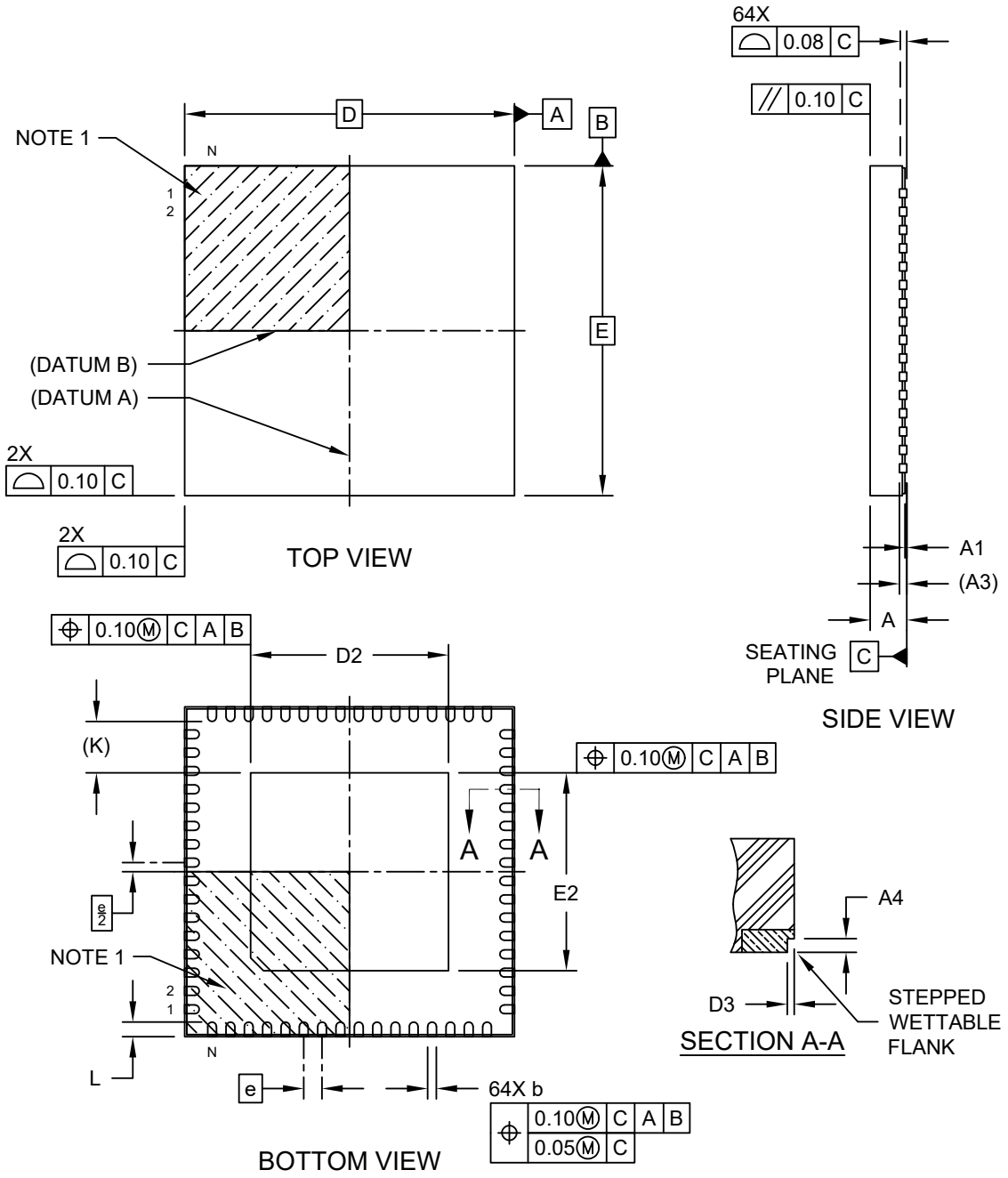
Microchip Technology Drawing C04-2514 Rev A


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.2.4. 64-pin VQFN with Wettable Flanks

**64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 mm Exposed Pad and Stepped Wettable Flanks**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|

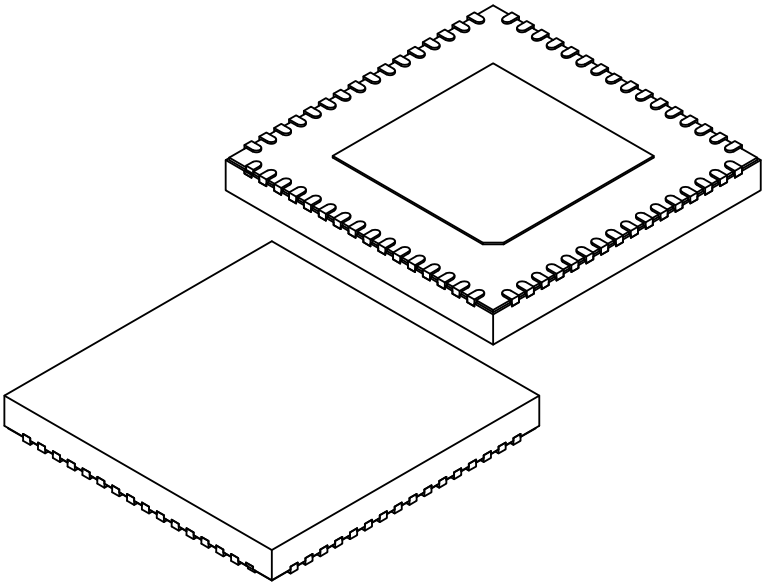



Microchip Technology Drawing C04-483-5LX Rev F Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 mm Exposed Pad and Stepped Wettable Flanks

**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 | | 64 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | 0.80 | 0.90 | 1.00 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Terminal Thickness | A3 | | 0.203 REF | |
| Overall Length | D | | 9.00 BSC | |
| Exposed Pad Length | D2 | 5.30 | 5.40 | 5.50 |
| Overall Width | E | | 9.00 BSC | |
| Exposed Pad Width | E2 | 5.30 | 5.40 | 5.50 |
| Terminal Width | b | 0.20 | 0.25 | 0.30 |
| Terminal Length | L | 0.30 | 0.40 | 0.50 |
| Terminal-to-Exposed-Pad | K | | 1.40 REF | |
| Wettable Flank Step Length | D3 | 0.035 | 0.060 | 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.

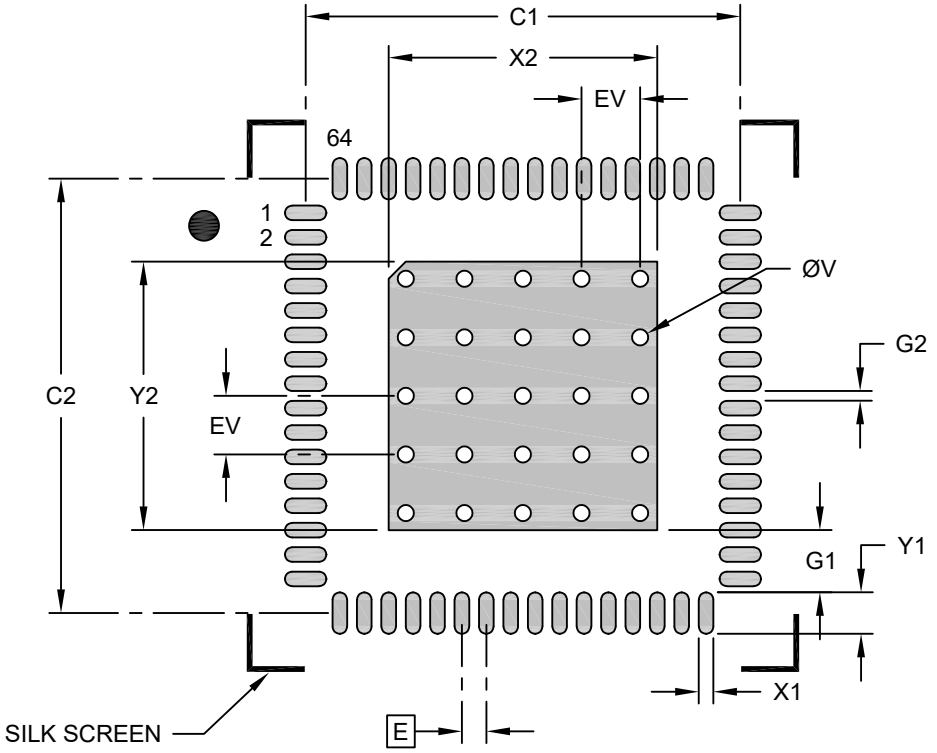
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-483-5LX Rev F Sheet 2 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 64-Lead Very Thin Plastic Quad Flat, No Lead Package (5LX) - 9x9x1.0 mm Body [VQFN] With 5.4 mm Exposed Pad and Stepped Wettable Flanks

**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 | | | 5.50 |
| Optional Center Pad Length | Y2 | | | 5.50 |
| 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 | 1.28 | | |
| 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

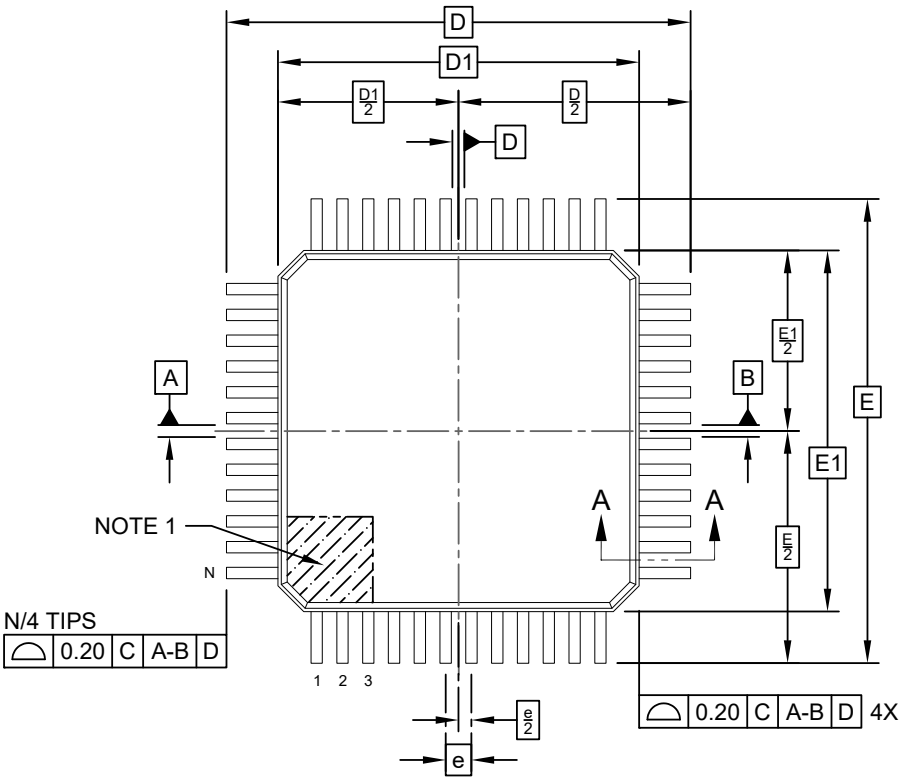
Microchip Technology Drawing C04-2483-5LX Rev F


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

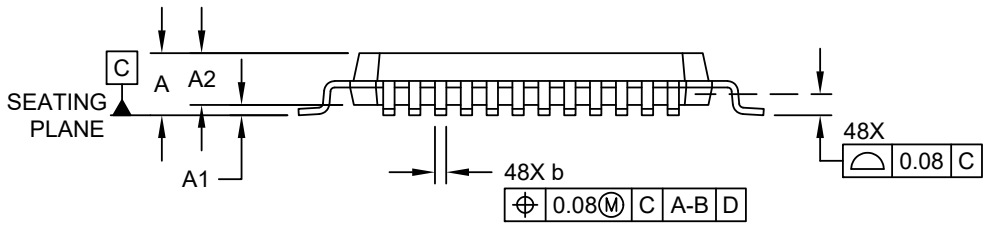
### 42.2.5. 48-Pin TQFP

**48-Lead Plastic Thin Quad Flatpack (Y8X) - 7x7x1.0 mm Body [TQFP]**

| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|




TOP VIEW



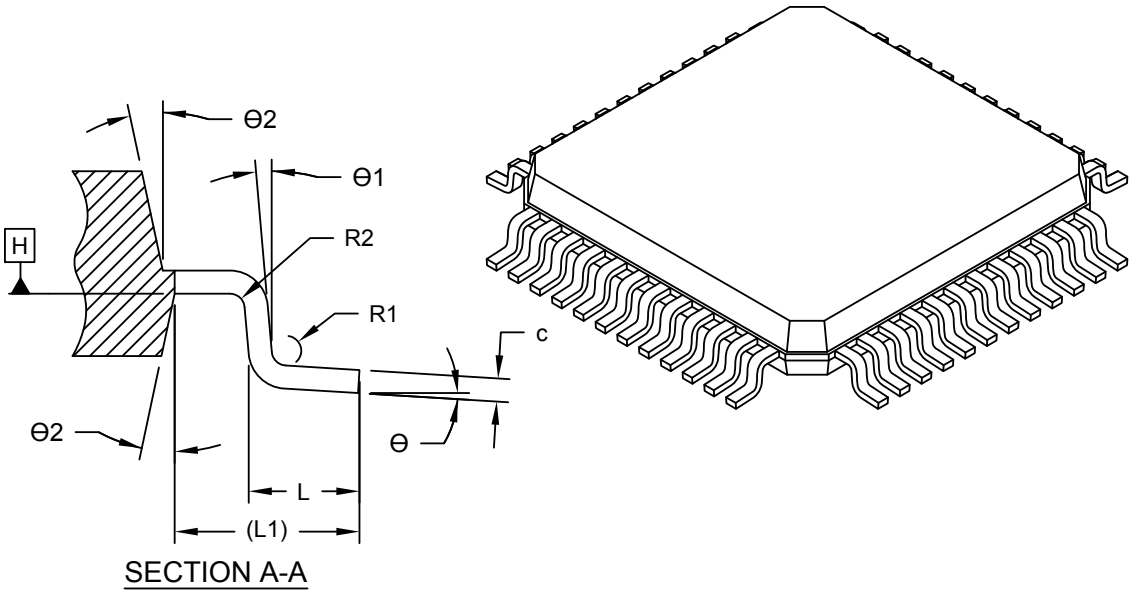

SIDE VIEW

Microchip Technology Drawing C04-00300-Y8X Rev E Sheet 1 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48-Lead Plastic Thin Quad Flatpack (Y8X) - 7x7x1.0 mm Body [TQFP]

**Note:** For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging




SECTION A-A

| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Terminals | N | | 48 | |
| Pitch | e | | 0.50 BSC | |
| Overall Height | A | - | - | 1.20 |
| Standoff | A1 | 0.05 | - | 0.15 |
| Molded Package Thickness | A2 | 0.95 | 1.00 | 1.05 |
| Overall Length | D | | 9.00 BSC | |
| Molded Package Length | D1 | | 7.00 BSC | |
| Overall Width | E | | 9.00 BSC | |
| Molded Package Width | E1 | | 7.00 BSC | |
| Terminal Width | b | 0.17 | 0.22 | 0.27 |
| Terminal Thickness | c | 0.09 | - | 0.16 |
| Terminal Length | L | 0.45 | 0.60 | 0.75 |
| Footprint | L1 | | 1.00 REF | |
| Lead Bend Radius | R1 | 0.08 | - | - |
| Lead Bend Radius | R2 | 0.08 | - | 0.20 |
| Foot Angle | Θ | 0° | 3.5° | 7° |
| Lead Angle | Θ1 | 0° | - | - |
| Mold Draft Angle | Θ2 | 11° | 12° | 13° |

Notes:

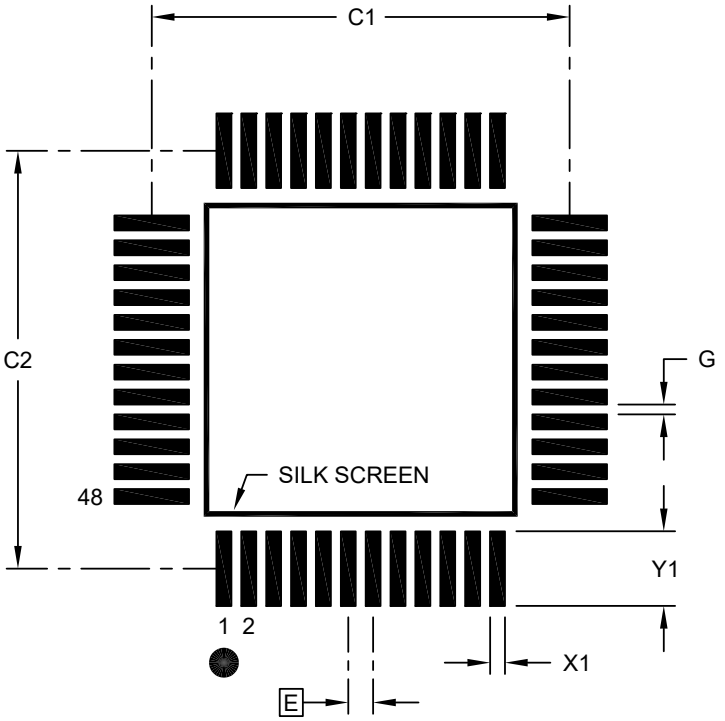
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-300-Y8X Rev D Sheet 2 of 2


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 48-Lead Plastic Thin Quad Flatpack (Y8X) - 7x7x1.0 mm Body [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 | | 8.40 | |
| Contact Pad Spacing | C2 | | 8.40 | |
| Contact Pad Width (X48) | X1 | | | 0.30 |
| Contact Pad Length (X48) | Y1 | | | 1.50 |
| Distance Between Pads | G | 0.20 | | |

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M.
   BSC: Basic Dimension. The theoretically exact value is shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during the reflow process.

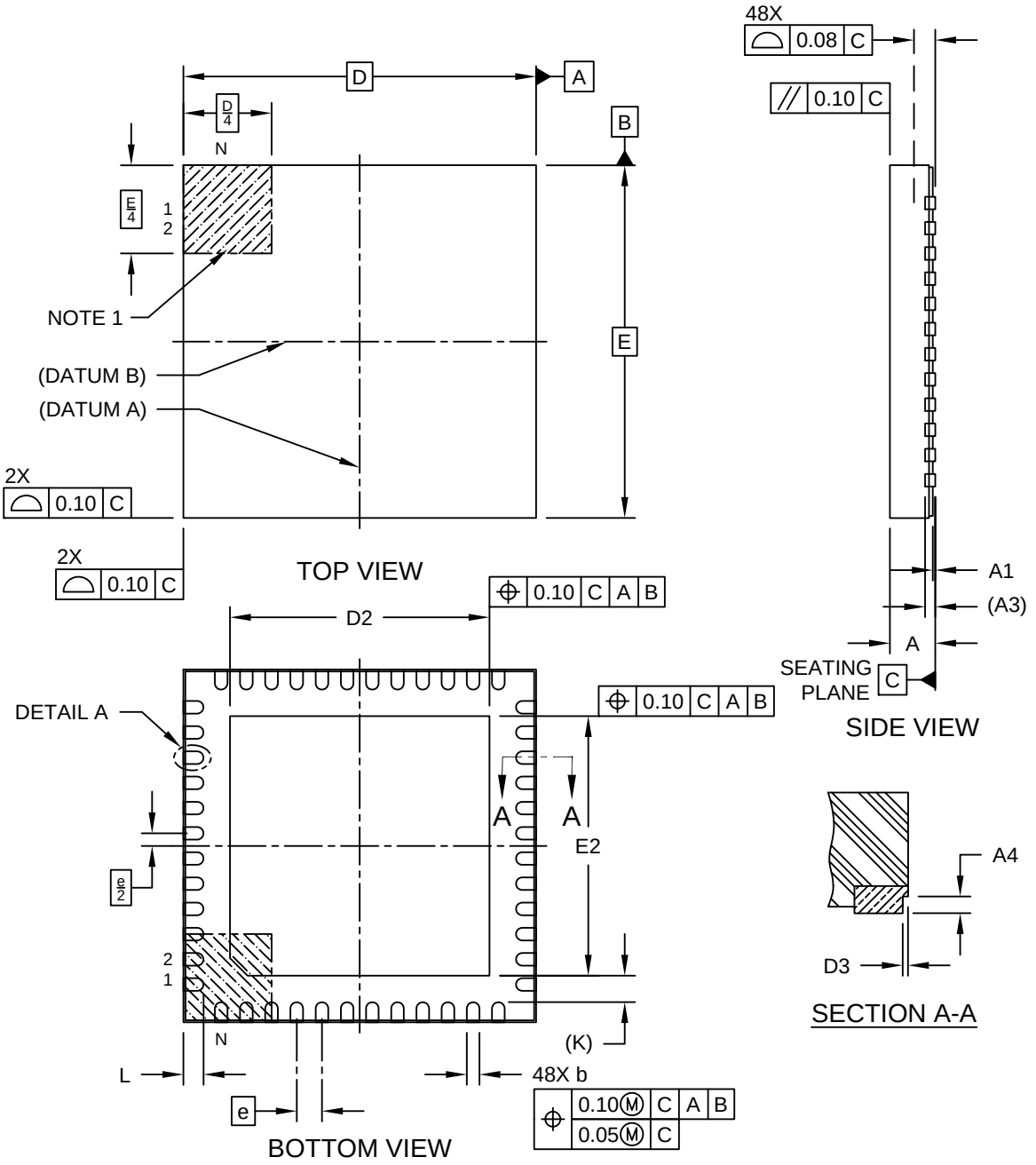
Microchip Technology Drawing C04-02300-Y8X Rev E


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

### 42.2.6. 48-Pin VQFN

**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 |
|---|---|

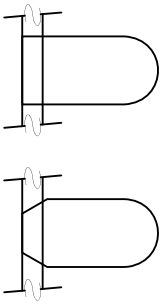



Microchip Technology Drawing C04-21493 Rev A Sheet 1 of 2

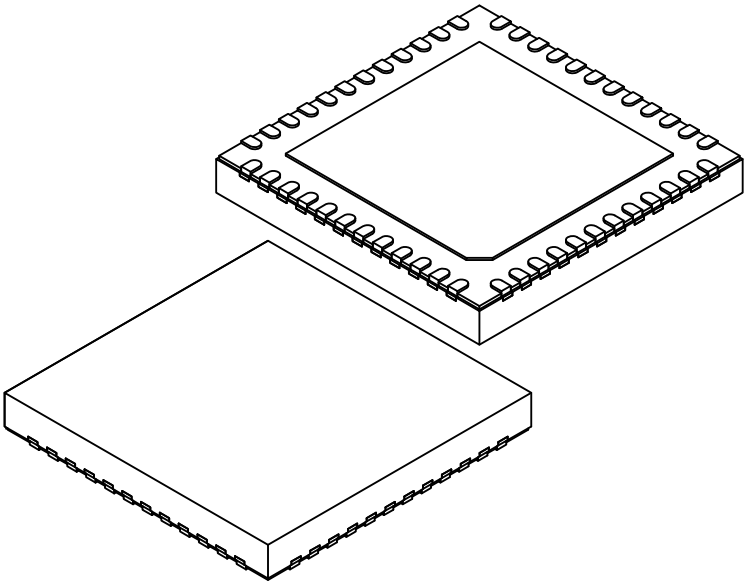

© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

## 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:

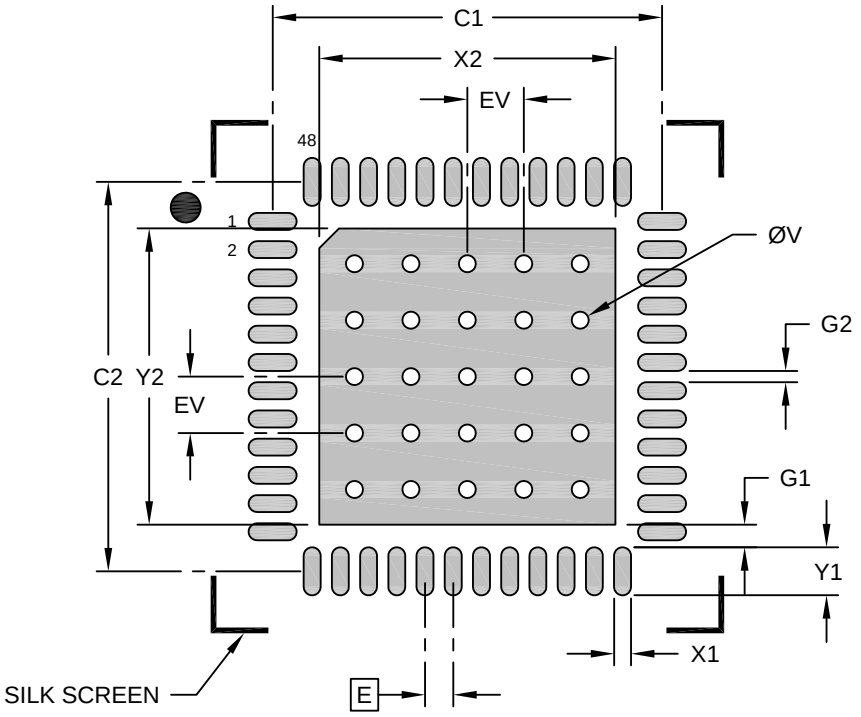
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information 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

## 42.3. Soldering Profile

The following table provides the recommended soldering profile from J-STD-20.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 42-1.** 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. |

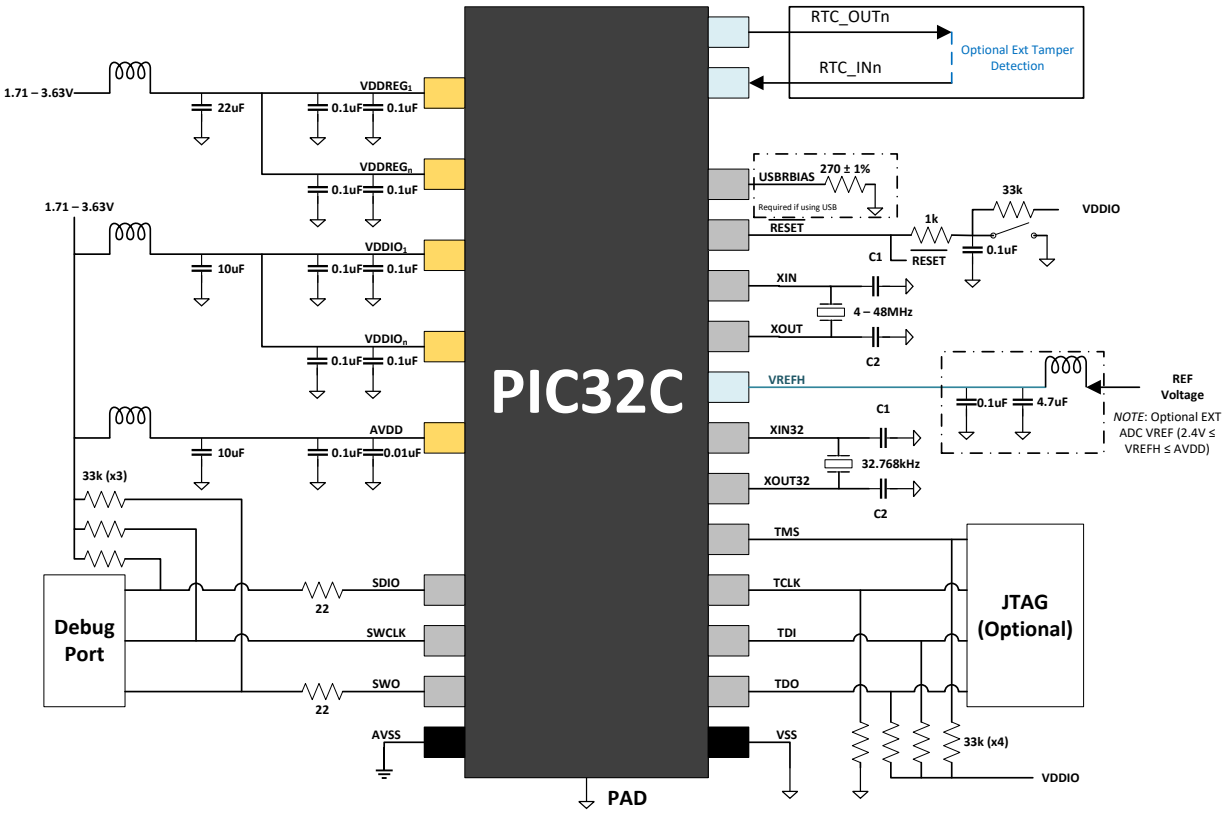
A maximum of three reflow passes is allowed per component.


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 43. Schematic Checklist

## 43.1. Introduction

This chapter describes a common checklist which should be used when starting and reviewing the schematics for a PIC32CM GC00 design. This chapter illustrates the recommended power supply connections, how to connect the external analog references, programmer, and debugger.





© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 44. Common Conventions

## 44.1. Numerical Notation

**Table 44-1.** Numerical Notation

| Symbol | Description |
|---|---|
| 165 | Decimal number |
| 0b0101 | Binary number (example 0b0101 = 5 decimal) |
| '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 |

## 44.2. Memory Size and Type

**Table 44-2.** 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) |
| 1 kbit/s | 1,000 bit/s rate (not 1,024 bit/s) |
| 1 Mbit/s | 1,000,000 bit/s rate |
| 1 Gbit/s | 1,000,000,000 bit/s rate |
| word | 32 bit |
| half-word | 16 bit |

## 44.3. Frequency and Time

**Table 44-3.** Frequency and Time

| Symbol | Description |
|---|---|
| kHz | 1 kHz = $10^3$ Hz = 1,000 Hz |
| KHz | 1 KHz = 1,024 Hz, 32 KHz = 32,768 Hz |
| MHz | 1 MHz = $10^6$ Hz = 1,000,000 Hz |
| GHz | 1 GHz = $10^9$ Hz = 1,000,000,000 Hz |
| s | second |
| ms | millisecond |
| µs | microsecond |
| ns | nanosecond |

## 44.4. Registers and Bits

**Table 44-4.** 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. |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 44-4.** Register and Bit Mnemonics (continued)

| Symbol | Description |
|---|---|
| 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) |
| 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.<br>Do not write any value to reserved bits of a fuse. |
| 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-on 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.

# 45. Acronyms and Abbreviations

The table below provides acronyms and abbreviations used in this document.

**Table 45-1.** 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 |
| DMAC | DMA (Direct Memory Access) Controller |
| DSU | Device Service Unit |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| EIC | External Interrupt Controller |
| EVSYS | Event System |
| FREQM | Frequency Meter |
| GCLK | Generic Clock Controller |
| 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 |
| MTB | Micro Trace Buffer |
| NMI | Non-maskable Interrupt |
| NVIC | Nested Vector Interrupt Controller |
| NVM | Nonvolatile Memory |
| NVMCTRL | Nonvolatile Memory Controller |
| OSC | Oscillator |
| PAC | Peripheral Access Controller |
| PC | Program Counter |
| PER | Period |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

**Table 45-1.** Acronyms and Abbreviations (continued)

| Abbreviation | Description |
|---|---|
| PLL | Phase Locked Loop |
| PM | Power Manager |
| POR | Power-on Reset |
| PORT | I/O Pin Controller |
| PWM | Pulse-Width Modulation |
| RAM | Random-Access Memory |
| REF | Reference |
| RTC | Real-Time Counter |
| RX | Receiver/Receive |
| SEEP | SmartEEPROM Page |
| SERCOM | Serial Communication Interface |
| SMBus | System Management Bus |
| SP | Stack Pointer |
| SPI | Serial Peripheral Interface |
| SRAM | Static Random Access Memory |
| SUPC | Supply Controller |
| SWD | Serial Wire Debug |
| TCC | Timer/Counter for Control |
| TRNG | True Random Number Generator |
| TX | Transmitter/Transmit |
| ULP | Ultra Low-Power |
| USART | Universal Synchronous and Asynchronous Serial Receiver and Transmitter |
| $V_{DD}$ | Common voltage to be applied to VDDIO and AVDD |
| $V_{DDIN}$ | Digital Supply Voltage |
| $V_{DDIO}$ | Digital Supply Voltage |
| $AV_{DD}$ | Analog Supply Voltage |
| VREF | Voltage Reference |
| WDT | Watchdog Timer |
| XOSC | Crystal Oscillator |


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.

# 46. Revision History

**Revision A - March 2026**

This is the initial release of this 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: 979-8-3371-2943-3

## 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

PIC32CM5112GC00048, PIC32CM5112GC00064, PIC32CM5112GC00100


© 2026 Microchip Technology Inc. and its subsidiaries
Downloaded from Arrow.com.