TOSHIBA

8 Bit Microcontroller
TLCS-870/C1 Series

TMP89FW24A

TOSHIBA CORPORATION
Semiconductor & Storage Products Company

Downloaded from Arrow.com.

©2026 TOSHIBA CORPORATION
All Rights Reserved

Downloaded from Arrow.com.

Revision History

Date	Revision	Comment
2012/5/18	1	・First Release
2013/2/15	2	・Contents Revised 2. CPU Core Add a note about constraints on SLEEP0/1 mode transition 3. Interrupt Control Circuit Delete unnecessary note 8. I/O Ports Correct the Rf value of P00/P01 24. Flash Memory Added note when doing erase/write Correct example code Added note about BOOTROM API calls to OCDE stop operation 26. Serial PROM Mode Add a note when a checksum error occurs 29. Electrical Characteristics Correct the erase / write / serial PROIM mode temperature range Delete the description about the solder ability test conditions

Content details (TMP89FW24A)

Date	Revision	Page	Update content
2012/5/18	1	—	・First Release
2013/2/15	2	34, 35	Add note to (5) SLEEP1 mode and (6) SLEEP0 mode
		41, 43	Add note to 2.3.6.2 IDLE1/2 and SLEEP1 modes, 2.3.6.3 IDLE0 and SLEEP0 modes
		58	Remove Note 1 from the interrupt sources table
		105	Correct the value of Rf in Figure 8-2 Note2 (1.2M to 500k)
		401	Add a note when using the Write, Erase and Security command Add a note when using the API of BOOTROM
		403	add a note 6 in Flash memory status register
		409 412 415 416 417 418	24.2.5 Monitoring a Ready/Busy state of the flash memory (FLSCRM<BUSY>) 24.3.1 Page Program 24.3.2 Sector Erase 24.3.3 Chip Erase (all erase) 24.3.4 Security Program 24.3.5 Security Erase Add a notes after confirming the BUSY flag
		420	24.4 Access to the Flash Memory Area Add a Note 5
		420, 421, 422,423	24.4.1.1 How to transfer and write a control program to the RAM area in RAM loader mode of the serial PROM mode Add a step 10, Add to wait 200us processing program, Delete Note2
		424, 427	24.4.2.1 How to write to the flash memory by transferring a control program to the RAM area Divided into four steps 4 and 5, Add step 10 and 17 (Wait 200us) Add to wait 200us processing program, Delete Note 2
		428 to 432	24.4.2 How to rewrite the flash memory by using a support program (API) of BOOTROM Changes to the program that run on the RAM without using .BTwrite
		433 to 435	24.4.2.3 How to set the security program by using a support program (API) of BOOTROM Changes to the program that run on the RAM
		436 to 438	24.4.2.4 How to rewrite the program itself by using the shadow RAM and the support program (API) of the BOOTROM Change to write programs that do not use the API
		439, 440	24.5 API Add Note 1, Note 2and Note 4、 Delete .BTWrite on List of API Delete 24.5.1 .BTWrite
		465	1. Flash memory erase command, 2. Flash memory write command Add a note when a checksum error occurs
		519	29.1 Absolute Maximum Ratings Added at the temperature range of programming, erase and serial PROM mode (0 to 70)
		520	29.2.1 MCU mode (Flash Programming or erasing) Change in the temperature range 0 to 70
		522	29.2.3 Serial PROM mode Change in the temperature range 0 to 70
		524	Add Note 5
		529, 530	29.9.1 MCU mode (Flash programming or erasing), 29.9.3 Serial PROM mode, 29.10.1 Write characteristics Change in the temperature range 0 to 70
		532	29.12 Handling Precaution Delete the description about the solder ability test conditions

Downloaded from Arrow.com.

Downloaded from Arrow.com.

February 2013

Dear Customers

Restrictions on the use of Flash memory in TMP89FW2x

This is to inform you of restrictions on the use of Flash memory in TMP89FW2x. If you have any questions or require any further information, please contact your local Toshiba sales office.

1. Product names

TMP89FW20UG、TMP89FW24FG、TMP89FW24DFG
TMP89FW20AUG、TMP89FW24AFG、TMP89FW24ADFG

2. The conditions under which the restrictions apply

(1) Writing, erasing, security setting or security erasing of Flash memory
(2) SLEEP0 mode or SLEEP1 mode are used.
(3) Writing or erasing operation on Flash memory in the serial PROM mode.
(4) Performing write operation by calling the API in the Boot ROM.
(5) Performing erase or setting/erasing security by calling the API in the Boot ROM.

3. Restrictions

Please read the instructions carefully before using these products.

(1) Following operations must be performed on the RAM or Shadow RAM.

・Performing write or erase operation on Flash memory.
・Setting or erasing the security

Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.
Note that in these processes interrupts must be prohibited; interrupt vectors and routines must be placed on RAM or Shadow RAM.

Applicable commands : Page Program
Chip Erase, Sector Erase
Security Program, Security Erase

(2) A transition to SLEEP0 mode or SLEEP1 mode must be performed on the RAM or the Shadow RAM. Note that in these processes interrupts must be prohibited; interrupt vectors and routines must be placed on RAM or Shadow RAM.

(3) If checksum errors occur when performing write/erase operation on the Flash memory in serial PROM mode, recheck the checksum using the Flash memory SUM output command.

(4) When performing write operation on the Flash memory, do not use the API for writing to the Flash memory in the BOOT ROM. Provide a write routine followed with the restrictions written in (1).

Applicable API : 0x1010 (.BTWrite) Do not use this API.

(5) When sector erasing, chip erasing, security setting or security releasing is performed by API on the BOOT ROM, hold wait time for 200μs or more before reading data from Flash memory or starting instruction fetch after the completion of calling API
Note that calling API and holding wait time must be performed on RAM or Shadow RAM.

Applicable API : 0x1012 (.BTEraseSec), 0x1014 (.BTEraseChip)
0x1018 (.BTSetSP), 0x1020 (.BTErsSP)

Downloaded from Arrow.com.

4. Assumed operations without introducing restrictions

(1) Read errors or malfunction of Flash memory
If Flash memory is accessed without holding wait time after performing write/erase operation, setting or erasing security of Flash memory, data read errors or undefined instructions may be generated.

(2) Malfunction in SLEEP0 mode or SLEEP1 mode
If a transition is performed in SLEEP0 mode or SLEEP1 mode on the Flash memory, the instruction may be undefined after Flash memory is returned to SLOW1 mode.

(3) Checksum errors in the serial PROM mode
If writing or erasing of Flash memory command is performed in the serial PROM mode, checksum error data may be returned. In this case, provisions would be different by depending on your Flash memory writing system. Check your system respectively.

(4) Write verify errors by calling API on BOOT ROM
If Flash memory is written by calling .BTWrite, verify errors may occur.

(5) Malfunction caused by reading Flash after erasing, security setting or releasing of Flash memory by calling API on the BOOTROM.
If Flash memory is accessed without holding wait time after performing erase operation, setting security or releasing of Flash memory, data read errors or undefined instructions may be generated.

Downloaded from Arrow.com.

Table of Contents

TMP89FW24A

1.1 Features.....1
1.2 Pin Assignment.....4
1.3 Block Diagram.....6
1.4 Pin Names and Functions.....7

2. CPU Core

2.1 Configuration.....13
2.2 Memory space.....13
2.2.1 Code area.....13
2.2.1.1 RAM
2.2.1.2 BOOTROM
2.2.1.3 Flash
2.2.2 Data area.....17
2.2.2.1 SFR
2.2.2.2 RAM
2.2.2.3 BOOTROM
2.2.2.4 Flash
2.3 System clock controller.....19
2.3.1 Configuration.....19
2.3.2 Control.....19
2.3.3 Functions.....22
2.3.3.1 Clock generator
2.3.3.2 Clock gear
2.3.3.3 Timing generator
2.3.4 Warm-up counter.....29
2.3.4.1 Warm-up counter operation when the oscillation is enabled by the hardware
2.3.4.2 Warm-up counter operation when the oscillation is enabled by the software
2.3.5 Operation mode control circuit.....31
2.3.5.1 Single-clock mode
2.3.5.2 Dual-clock mode
2.3.5.3 STOP mode
2.3.5.4 Transition of operation modes
2.3.6 Operation Mode Control.....37
2.3.6.1 STOP mode
2.3.6.2 IDLE1/2 and SLEEP1 modes
2.3.6.3 IDLE0 and SLEEP0 modes
2.3.6.4 SLOW mode
2.4 Reset Control Circuit.....49
2.4.1 Configuration.....49
2.4.2 Control.....49
2.4.3 Functions.....51
2.4.4 Reset Signal Generating Factors.....53
2.4.4.1 Power-on reset
2.4.4.2 External reset input (RESET pin input)
2.4.4.3 Voltage detection reset
2.4.4.4 Watchdog timer reset
2.4.4.5 System clock reset
2.4.4.6 Internal factor reset detection status register
2.4.4.7 How to use the external reset input pin as a port

Downloaded from Arrow.com.

3. Interrupt Control Circuit

3.1 Configuration.....59
3.2 Interrupt Latches (IL30 to IL3).....60
3.3 Interrupt Enable Register (EIR).....61
3.3.1 Interrupt master enable flag (IMF).....61
3.3.2 Individual interrupt enable flags (EF30 to EF4).....61
3.4 Maskable Interrupt Priority Change Function.....64
3.5 Interrupt Sequence.....66
3.5.1 Initial Setting.....66
3.5.2 Interrupt acceptance processing.....66
3.5.3 Saving/restoring general-purpose registers.....68
3.5.3.1 Using PUSH and POP instructions
3.5.3.2 Using data transfer instructions
3.5.3.3 Using a register bank to save/restore general-purpose registers
3.5.4 Interrupt return.....70
3.6 Software Interrupt (INTSW).....71
3.6.1 Address error detection.....71
3.6.2 Debugging.....71
3.7 Undefined Instruction Interrupt (INTUNDEF).....71

4. External Interrupt control circuit

4.1 Configuration.....73
4.2 Control.....73
4.3 Function.....76
4.3.1 Low power consumption function.....77
4.3.2 External interrupt 0.....77
4.3.3 External interrupts 1/2/3.....78
4.3.3.1 Interrupt request signal generating condition detection function
4.3.3.2 A noise canceller pass signal monitoring function when interrupt request signals are generated
4.3.3.3 Noise cancel time selection function
4.3.4 External interrupt 5.....79

5. Watchdog Timer (WDT)

5.1 Configuration.....81
5.2 Control.....82
5.3 Functions.....84
5.3.1 Setting of enabling/disabling the watchdog timer operation.....84
5.3.2 Setting the clear time of the 8-bit up counter.....84
5.3.3 Setting the overflow time of the 8-bit up counter.....85
5.3.4 Setting an overflow detection signal of the 8-bit up counter.....85
5.3.5 Writing the watchdog timer control codes.....86
5.3.6 Reading the 8-bit up counter.....86
5.3.7 Reading the watchdog timer status.....86

6. Power-on Reset Circuit

6.1 Configuration.....89
6.2 Function.....89

Downloaded from Arrow.com.

7. Voltage Detection Circuit

7.1 Configuration....91
7.2 Control....92
7.3 Function....93
7.3.1 Enabling/disabling the voltage detection operation....93
7.3.2 Selecting the voltage detection operation mode....93
7.3.3 Selecting the detection voltage level....94
7.3.4 Voltage detection flag and voltage detection status flag....94
7.4 Register Settings....96
7.4.1 Setting procedure when the operation mode is set to generate INTVLTD interrupt request signals....96
7.4.2 Setting procedure when the operation mode is set to generate voltage detection reset signals....96
7.5 Caution in the using voltage detevtion circuit....97

8. IO Ports

8.1 I/O Port Control Registers....101
8.2 List of I/O Port Settings....102
8.3 I/O Port Registers....105
8.3.1 Port P0 (P03 ~ P00)....105
8.3.2 P1 (P13 ~ P10)....109
8.3.3 Port P2 (P25 ~ P20)....113
8.3.4 Port P3 (P37 ~ P30)....118
8.3.5 Port P4 (P47 ~ P40)....121
8.3.6 Port P5 (P57 ~ P50)....124
8.3.7 Port P6 (P67 ~ P60)....126
8.3.8 Port P7 (P77 ~ P70)....128
8.3.9 Port P8 (P87 ~ P80)....130
8.3.10 Port P9 (P97 ~ P90)....133
8.4 Peripheral input/output select function....136

9. Special Function Registers

9.1 SFR1 (0x00000 to 0x0003F)....141
9.2 SFR2 (0x00F00 to 0x00FFF)....142
9.3 SFR3 (0x00E40 to 0x00EFF)....144

10. Low Power Consumption Function for Peripherals

10.1 Control....148

11. Divider Output ($\overline{\text{DVO}}$)

11.1 Configuration....151
11.2 Control....152
11.3 Function....153

Downloaded from Arrow.com.

12. Time Base Timer (TBT)

12.1 Time Base Timer 155
12.1.1 Configuration 155
12.1.2 Control 155
12.1.3 Functions 156

13. 16-bit Timer Counter (TCA)

13.1 Configuration 160
13.2 Control 161
13.3 Low Power Consumption Function 166
13.4 Timer Function 167
13.4.1 Timer mode 167
13.4.1.1 Setting
13.4.1.2 Operation
13.4.1.3 Auto capture
13.4.1.4 Register buffer configuration
13.4.2 External trigger timer mode 171
13.4.2.1 Setting
13.4.2.2 Operation
13.4.2.3 Auto capture
13.4.2.4 Register buffer configuration
13.4.3 Event counter mode 173
13.4.3.1 Setting
13.4.3.2 Operation
13.4.3.3 Auto capture
13.4.3.4 Register buffer configuration
13.4.4 Window mode 175
13.4.4.1 Setting
13.4.4.2 Operation
13.4.4.3 Auto capture
13.4.4.4 Register buffer configuration
13.4.5 Pulse width measurement mode 177
13.4.5.1 Setting
13.4.5.2 Operation
13.4.5.3 Capture process
13.4.6 Programmable pulse generate (PPG) mode 180
13.4.6.1 Setting
13.4.6.2 Operation
13.4.6.3 Register buffer configuration
13.5 Noise Canceller 183
13.5.1 Setting 183

14. 16-bit Timer Counter (TCB)

14.1 Configuration 186
14.2 Control 187
14.3 Low Power Consumption Function 191
14.4 Timer Function 192
14.4.1 Timer mode 192
14.4.1.1 Setting
14.4.1.2 Operation
14.4.1.3 Auto capture
14.4.1.4 Register buffer configuration
14.4.2 External trigger timer mode 196
14.4.2.1 Setting
14.4.2.2 Operation
14.4.2.3 Auto capture
14.4.2.4 Register buffer configuration

Downloaded from Arrow.com.

14.4.3 Event counter mode....198
14.4.3.1 Setting
14.4.3.2 Operation
14.4.3.3 Auto capture
14.4.3.4 Register buffer configuration
14.4.4 Window mode....200
14.4.4.1 Setting
14.4.4.2 Operation
14.4.4.3 Auto capture
14.4.4.4 Register buffer configuration
14.4.5 Pulse width measurement mode....202
14.4.5.1 Setting
14.4.5.2 Operation
14.4.5.3 Capture process
14.4.6 Programmable pulse generate (PPG) mode....205
14.4.6.1 Setting
14.4.6.2 Operation
14.4.6.3 Register buffer configuration
14.5 Noise Canceller....208
14.5.1 Setting....208

15. 10-Bit Timer/Counter (TCC)

15.1 Configuration....210
15.2 Control....211
15.3 Low Power Consumption Function....217
15.4 Configuring Control and Data Registers....218
15.5 Features....220
15.5.1 Programmable pulse generator output (PPG output)....220
15.5.1.1 50% duty mode
15.5.1.2 Variable duty mode
15.5.1.3 PPGC01/PPGC02 independent mode
15.5.2 Starting a count....224
15.5.2.1 Command start and capture mode(TC0CR2<CSTC>="00")
15.5.2.2 Command start and trigger start mode(TC0CR2<CSTC>="01")
15.5.2.3 Trigger start mode(TC0CR2<CSTC>="10")
15.5.3 Trigger capture....227
15.5.3.1 Description
15.5.3.2 Register settings
15.5.4 Trigger start/stop acceptance mode....228
15.5.4.1 Selecting an input signal logic for the TCC0 pin (Trigger input)
15.5.4.2 Specifying whether triggers are always accepted or ignored when PPG outputs are active
15.5.4.3 Ignoring triggers when PPG outputs are active
15.5.5 Configuring how the timer stops....230
15.5.5.1 Counting stopped with the outputs initialized
15.5.5.2 Counting stopped with the outputs maintained
15.5.5.3 Counting stopped with the outputs initialized at the end of the period
15.5.6 One - time/continuous output mode....231
15.5.6.1 One-time output mode
15.5.6.2 Continuous output mode
15.5.7 PPG output control (Initial value/output logic, enabling/disabling output)....233
15.5.7.1 Specifying initial values and output logic for PPG outputs
15.5.7.2 Enabling or disabling PPG outputs
15.5.7.3 Using the TCC0 as a normal timer/counter
15.5.8 Eliminating noise from the TCC0 pin input....234
15.5.9 Interrupts....235
15.5.9.1 INTTCC0T (Trigger start interrupt)
15.5.9.2 INTTCC0P (Period interrupt)
15.5.9.3 INTEMG0 (Emergency output stop interrupt)
15.5.10 Emergency PPG output stop feature....237
15.5.10.1 Enabling/disabling input on the EMG0 pin
15.5.10.2 Monitoring the emergency PPG output stop state
15.5.10.3 EMG interrupt
15.5.10.4 Canceling the emergency PPG output stop state
15.5.10.5 Restarting the timer after canceling the emergency PPG output stop state
15.5.10.6 Response time between EMG0 pin input and PPG outputs being initialized
15.5.11 TCC0 operation and microcontroller operating mode....238
15.5.12 Considerations for Using the development tools of TCC0....239

Downloaded from Arrow.com.

16. 8-bit Timer Counter (TC0)

16.1 Configuration.....242
16.2 Control.....243
16.2.1 Timer counter 00.....243
16.2.2 Timer counter 01.....245
16.2.3 Common to timer counters 00 and 01.....247
16.2.4 Operation modes and usable source clocks.....249
16.3 Low Power Consumption Function.....250
16.4 Functions.....251
16.4.1 8-bit timer mode.....251
16.4.1.1 Setting
16.4.1.2 Operation
16.4.1.3 Double buffer
16.4.2 8-bit event counter mode.....254
16.4.2.1 Setting
16.4.2.2 Operation
16.4.2.3 Double buffer
16.4.3 8-bit pulse width modulation (PWM) output mode.....256
16.4.3.1 Setting
16.4.3.2 Operations
16.4.3.3 Double buffer
16.4.4 8-bit programmable pulse generate (PPG) output mode.....261
16.4.4.1 Setting
16.4.4.2 Operation
16.4.4.3 Double buffer
16.4.5 16-bit timer mode.....265
16.4.5.1 Setting
16.4.5.2 Operations
16.4.5.3 Double buffer
16.4.6 16-bit event counter mode.....269
16.4.6.1 Setting
16.4.6.2 Operations
16.4.6.3 Double buffer
16.4.7 12-bit pulse width modulation (PWM) output mode.....271
16.4.7.1 Setting
16.4.7.2 Operations
16.4.7.3 Double buffer
16.4.8 16-bit programmable pulse generate (PPG) output mode.....277
16.4.8.1 Setting
16.4.8.2 Operations
16.4.8.3 Double buffer

17. Real Time Clock (RTC)

17.1 Configuration.....281
17.2 Control.....281
17.3 Function.....282
17.3.1 Low Power Consumption Function.....282
17.3.2 Enabling/disabling the real time clock operation.....282
17.3.3 Selecting the interrupt generation interval.....282
17.4 Real Time Clock Operation.....283
17.4.1 Enabling the real time clock operation.....283
17.4.2 Disabling the real time clock operation.....283

18. Asynchronous Serial Interface (UART)

18.1 Configuration.....286
18.2 Control.....287

Downloaded from Arrow.com.

18.3 Low Power Consumption Function.....291
18.4 Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed 292
18.5 Activation of STOP, IDLE0 or SLEEP0 Mode.....293
18.5.1 Transition of register status.....293
18.5.2 Transition of TXD pin status.....293
18.6 Transfer Data Format.....294
18.7 Infrared Data Format Transfer Mode.....294
18.8 Transfer Baud Rate.....295
18.8.1 Transfer baud rate calculation method.....296
18.8.1.1 Bit width adjustment using UART0CR2<RTSEL>
18.8.1.2 Calculation of set values of UART0CR2<RTSEL> and UART0DR
18.9 Data Sampling Method.....299
18.10 Received Data Noise Rejection.....301
18.11 Transmit/Receive Operation.....302
18.11.1 Data transmit operation.....302
18.11.2 Data receive operation.....302
18.12 Status Flag.....303
18.12.1 Parity error.....303
18.12.2 Framing Error.....304
18.12.3 Overrun error.....305
18.12.4 Receive Data Buffer Full.....308
18.12.5 Transmit busy flag.....309
18.12.6 Transmit Buffer Full.....309
18.13 Receiving Process.....310
18.14 AC Properties.....312
18.14.1 IrDA properties.....312

19. Synchronous Serial Interface (SIO)

19.1 Configuration.....314
19.2 Control.....315
19.3 Low Power Consumption Function.....318
19.4 Functions.....319
19.4.1 Transfer format.....319
19.4.2 Serial clock.....319
19.4.3 Transfer edge selection.....319
19.5 Transfer Modes.....321
19.5.1 8-bit transmit mode.....321
19.5.1.1 Setting
19.5.1.2 Starting the transmit operation
19.5.1.3 Transmit buffer and shift operation
19.5.1.4 Operation on completion of transmission
19.5.1.5 Stopping the transmit operation
19.5.2 8-bit Receive Mode.....326
19.5.2.1 Setting
19.5.2.2 Starting the receive operation
19.5.2.3 Operation on completion of reception
19.5.2.4 Stopping the receive operation
19.5.3 8-bit transmit/receive mode.....330
19.5.3.1 Setting
19.5.3.2 Starting the transmit/receive operation
19.5.3.3 Transmit buffer and shift operation
19.5.3.4 Operation on completion of transmission/reception
19.5.3.5 Stopping the transmit/receive operation
19.6 AC Characteristics.....335

Downloaded from Arrow.com.

20. Serial Bus Interface (SBI)

20.1 Communication Format.....338
20.1.1 I2C bus.....338
20.1.2 Free data format.....339
20.2 Configuration.....340
20.3 Control.....341
20.4 Functions.....344
20.4.1 Low Power Consumption Function.....344
20.4.2 Selecting the slave address match detection and the GENERAL CALL detection.....344
20.4.3 Selecting the number of clocks for data transfer and selecting the acknowledgement or non-acknowledgment mode 344
20.4.3.1 Number of clocks for data transfer
20.4.3.2 Output of an acknowledge signal
20.4.4 Serial clock.....346
20.4.4.1 Clock source
20.4.4.2 Clock synchronization
20.4.5 Master/slave selection.....348
20.4.6 Transmitter/receiver selection.....348
20.4.7 Start/stop condition generation.....349
20.4.8 Interrupt service request and release.....349
20.4.9 Setting of serial bus interface mode.....350
20.4.10 Software reset.....350
20.4.11 Arbitration lost detection monitor.....350
20.4.12 Slave address match detection monitor.....352
20.4.13 GENERAL CALL detection monitor.....352
20.4.14 Last received bit monitor.....353
20.4.15 Slave address and address recognition mode specification.....353
20.5 Data Transfer of I2C Bus.....354
20.5.1 Device initialization.....354
20.5.2 Start condition and slave address generation.....354
20.5.3 1-word data transfer.....355
20.5.3.1 When SBI0SR2<MST> is "1" (Master mode)
20.5.3.2 When SBI0SR2<MST> is "0" (Slave mode)
20.5.4 Stop condition generation.....359
20.5.5 Restart.....359
20.6 AC Specifications.....361

21. Key-on Wakeup (KWU)

21.1 Configuration.....363
21.2 Control.....364
21.3 Functions.....365

22. 10-bit AD Converter (ADC)

22.1 Configuration.....367
22.2 Control.....368
22.3 Functions.....372
22.3.1 Single mode.....372
22.3.2 Repeat mode.....372
22.3.3 AD operation disable and forced stop of AD operation.....373
22.4 Register Setting.....374
22.5 Starting STOP/IDLE0/SLOW Modes.....374
22.6 Analog Input Voltage and AD Conversion Result.....375

Downloaded from Arrow.com.

22.7 Precautions about the AD Converter.....376
22.7.1 Analog input pin voltage range.....376
22.7.2 Analog input pins used as input/output ports.....376
22.7.3 Noise countermeasure.....376

23. LCD Driver

23.1 Configuration.....378
23.2 Control.....379
23.3 Low Power Consumption Function.....382
23.4 Functions.....383
23.4.1 LCD Display Control (LCDCR1<EDSP>).....383
23.4.1.1 Operation at Reset
23.4.1.2 Operation in IDLE0, SLEEP0 and STOP modes
23.4.1.3 Operation in SLOW mode
23.4.1.4 Display operation according to the base frequency setting (LCDCR1<SLF>) (fail-safe)
23.4.1.5 Display operation according to the low power consumption register (fail-safe)
23.4.2 LCD Drive Methods (LCDCR1<DUTY>).....385
23.4.3 Frame Frequency (LCDCR1<SLF>).....386
23.4.4 Internal/External Bleeder Resistance Switching Control 387
23.4.5 Low Internal Bleeder Resistance Connection Time Selection (LCDCR2<LRSE>).....387
23.4.6 High Internal Bleeder Resistance Selection (LCDCR2<BRH>).....390
23.5 LCD Display Operation.....391
23.6 Display Data Setting.....392
23.7 Examples of How To Control the LCD Driver.....393
23.7.1 Initialization.....393
23.7.2 Display Data Setting.....393
23.7.3 Drive Output Examples.....396

24. Flash Memory

24.1 Flash Memory Control.....402
24.2 Functions.....404
24.2.1 Flash memory command sequence execution (FLSCR1<FLSMD>).....404
24.2.2 Flash memory area switching (FLSCR1<FAREA>).....405
24.2.3 RAM area switching (SYSCR3<RAREA>).....407
24.2.4 BOOTROM area switching (FLSCR1<BAREA>).....407
24.2.5 Monitoring a Ready / Busy state of the flash memory (FLSCRM<BUSY>).....409
24.2.6 Port input control register (SPCR<PIN0,PIN1>).....409
24.2.7 Shadow RAM mapping control in the data area (SDWCR1<DADIS>).....409
24.3 Command Sequence.....410
24.3.1 Page Program.....412
24.3.2 Sector Erase (partial erase in units of a Sector).....415
24.3.3 Chip Erase (all erase).....416
24.3.4 Security Program.....417
24.3.5 Security Erase.....418
24.3.6 Product ID Entry.....419
24.3.7 Product ID Exit.....419
24.4 Access to the Flash Memory Area.....420
24.4.1 Flash memory control in serial PROM mode.....420
24.4.1.1 How to transfer and write a control program to the RAM area in RAM loader mode of the serial PROM mode
24.4.2 Flash memory control in MCU mode.....424
24.4.2.1 How to write to the flash memory by transferring a control program to the RAM area
24.4.2.2 How to rewrite the flash memory by using a support program (API) of BOOTROM
24.4.2.3 How to set the security program by using a support program (API) of BOOTROM
24.4.2.4 How to rewrite the program itself by using the shadow RAM and the support program (API) of the BOOTROM
24.4.2.5 How to read data from flash memory
24.5 API (Application Programming Interface).....440
24.5.1 .BTRead.....441
24.5.2 .BTEraseSec.....441

Downloaded from Arrow.com.

24.5.3 .BTEraseChip....442
24.5.4 .BTGetSP....442
24.5.5 .BTSetSP....442
24.5.6 .BTErsSP....442
24.5.7 .BTConvAdr....442
24.5.8 .BTCalcUART....443
24.5.9 .BTUpdSD....444

25. Shadow RAM

25.1 Configuration....447
25.2 Control....448
25.3 Memory Map....450
25.4 Functions....451
25.4.1 Copying the Flash memory....451
25.4.2 Powering off the Flash memory (SDWCR1<FLSOFF>)....454
25.4.3 Shadow RAM mapping control in the data area (SDWCR1<DADIS>)....455
25.4.4 Shadow RAM mapping control (SDWCR1<SDWDIS>)....455
25.4.5 Shadow RAM write control in the data area (SDWCR1<DAWREN>)....455
25.4.6 Data RAM mapping control (SDWCR1<EXPRAM>)....455

26. Serial PROM Mode

26.1 Outline....457
26.2 Security....457
26.3 Serial PROM Mode Setting....458
26.3.1 Serial PROM mode control pins....458
26.4 Examples of Connections for On-Board Programming....460
26.5 Starting in Serial PROM Mode....461
26.6 Interface Specifications....462
26.6.1 SIO Communication....462
26.6.2 UART Communication....462
26.7 Memory Mapping....464
26.8 Operation Commands....465
26.8.1 Flash memory erase command (0xF0)....467
26.8.1.1 Specifying the erase area
26.8.2 Flash Memory Write Command (operation command: 0x30)....470
26.8.3 Flash memory read command (operation command: 0x40)....472
26.8.4 RAM loader command (operation command: 0x60)....474
26.8.5 Flash memory SUM output command (operation command: 0x90)....476
26.8.6 Product ID code output command (operation command: 0xC0)....477
26.8.7 Flash memory status output command (0xC3)....479
26.8.7.1 Flash memory status code
26.8.8 Flash memory security setting command (0xFA)....482
26.8.9 Clock change command (operation command: 0xA0)....484
26.9 Error Codes....485
26.10 Checksum (SUM)....486
26.10.1 Calculation method....486
26.10.2 Calculation data....486
26.11 Intel Hex Format (Binary)....487
26.12 Security....489
26.12.1 Password....489
26.12.1.1 How a password can be specified
26.12.1.2 Password structure
26.12.1.3 Password setting, clearance and authentication
26.12.1.4 Password values and setting range
26.12.2 Security Program....493
26.12.2.1 How Security Program works

Downloaded from Arrow.com.

26.12.2.2 Enabling or disabling Security Program
26.12.3 Option codes....494
26.12.4 Recommended settings....496
26.13 Flowchart....497
26.14 AC Characteristics (UART)....498
26.14.1 Reset timing....500
26.14.2 Flash memory erase command (0xF0)....500
26.14.3 Flash memory write command (0x30)....501
26.14.4 Flash memory read command (0x40)....501
26.14.5 RAM loader command (0x60)....502
26.14.6 Flash memory SUM output command (0x90)....502
26.14.7 Product ID code output command (0xC0)....502
26.14.8 Flash memory status output command (0xC3)....503
26.14.9 Flash memory security setting command (0xFA)....503
26.14.10 Clock change command (0xA0)....503
26.15 AC Characteristics (SIO)....504
26.15.1 SIO transfer timing....506
26.15.2 Reset timing....506
26.15.3 Flash memory erase command (0xF0)....507
26.15.4 Flash memory write command (0x30)....508
26.15.5 Flash memory read command (0x40)....509
26.15.6 RAM Loader Command (0x60)....510
26.15.7 Flash memory SUM output command (0x90)....511
26.15.8 Product ID code output command (0xC0)....511
26.15.9 Flash memory status output command (0xC3)....511
26.15.10 Flash memory security setting command (0xFA)....512

27. On-chip Debug Function (OCD)

27.1 Features....513
27.2 Control Pins....513
27.3 How to Connect the On-chip Debug Emulator to a Target System....515
27.4 Security....515

28. Input/Output Circuit

28.1 Control Pins....517

29. Electrical Characteristics

29.1 Absolute Maximum Ratings....519
29.2 Operating Conditions....520
29.2.1 MCU mode (Flash Programming or erasing)....520
29.2.2 MCU mode (Except Flash Programming or erasing)....521
29.2.3 Serial PROM mode....522
29.3 DC Characteristics....523
29.4 AD Conversion Characteristics....525
29.5 Power-on Reset Circuit Characteristics....526
29.6 Voltage Detecting Circuit Characteristics....527
29.7 16-bit Timer Counter(TCB) Characteristics....528
29.8 LCD Characteristics....528
29.9 AC Characteristics....529
29.9.1 MCU mode (Flash programming or erasing)....529
29.9.2 MCU mode (Except Flash Programming or erasing)....529
29.9.3 Serial PROM mode....530

Downloaded from Arrow.com.

29.10 Flash Characteristics 530
29.10.1 Write characteristics.....530
29.11 Recommended Oscillating Condition.....531
29.12 Handling Precaution.....532

30. Package Dimensions

31.

Downloaded from Arrow.com.

CMOS 8-Bit Microcontroller

TMP89FW24A

The TMP89FW24A is a single-chip 8-bit high-speed and high-functionality microcomputer incorporating 126976 bytes of Flash Memory.

Product No.	ROM (Flash)	RAM	Package	Emulation Chip
TMP89FW24AFG	126976 bytes	3072 bytes	LQFP80-P-1212-0.50F	-
TMP89FW24ADFG			QFP80-P-1420-0.80M	

1.1 Features

1. 8-bit single chip microcomputer TLCS-870/C1 series
   - Instruction execution time :
     62.5 ns (at 16 MHz)
     122 μs (at 32.768 kHz)
   - 133 types & 732 basic instructions

2. 30 interrupt sources (External : 5 Internal : 25 , Except reset)

3. Input / Output ports (68 pins)
   - Large current output: 11 pins (Typ. 10mA)

4. Input ports (1)

5. Watchdog timer
   - Interrupt or reset can be selected by the program.

6. Power-on reset circuit

7. Voltage detection circuit

8. Divider output function

9. Time base timer

10. 16-bit timer counter (TCA) : 1 ch
    - Timer, External trigger, Event Counter, Window, Pulse width measurement, PPG OUTPUT modes

11. 16-bit timer counter (TCB) : 1 ch
    - Timer, External trigger, Event Counter, Window, Pulse width measurement, PPG OUTPUT modes

12. 10-bit timer counter (TCC) : 1 ch (2 output pins)
    - 2ports output PPG (Programmed Pulse Generator)
    - Variable Duty output mode
    - 50%duty output mode

Downloaded from Arrow.com.

- External-triggered start and stop
- Emargency stop pin

13. 8-bit timer counter (TC0) : 4 ch
    - Timer, Event Counter, PWM, PPG OUTPUT modes
    - Usable as a 16-bit timer, 12-bit PWM output and 16-bit PPG output by the cascade connection of two channels.

14. Real time clock

15. UART : 3ch

16. SIO : 1ch Note : Two SIO channels can be used at the same time.

17. I^2C/SIO : 1ch

18. Key-on wake-up : 3 ch

19. 10-bit successive approximation type AD converter
    - Analog input : 8ch

20. LCD driver / controller
    - LCD direct drive capability (40 seg × 4 com)
    - 1/4, 1/3, 1/2 duties or static drive are programmably selectable
    - Internal bleeder resistance for LCD bias voltage (Usable as an external bleeder resistance by external bleeder resistance connection pins)

21. Shadow RAM

22. On-chip debug function
    - Break/Event
    - Trace
    - RAM monitor
    - Flash memory writing

23. Internal high-frequency clock oscillation circuit (Typ. 10 MHz)

    System clock is started by internal high frequency after reset.

24. Clock operation mode control circuit : 2 circuit

    Single clock mode / Dual clock mode

25. Low power consumption operation (8 mode)
    - STOP mode:

      Oscillation stops. (Battery/Capacitor back-up.)
    - SLOW1 mode:

      Low power consumption operation using low-frequency clock.(High-frequency clock stop.)
    - SLOW2 mode:

      Low power consumption operation using low-frequency clock.(High-frequency clock oscillate.)
    - IDLE0 mode:

      CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using high frequency clock. Released when the reference time set to TBT has elapsed.

Downloaded from Arrow.com.

- IDLE1 mode:

  The CPU stops, and peripherals operate using high frequency clock. Release by interruputs(CPU restarts).
- IDLE2 mode:

  CPU stops and peripherals operate using high and low frequency clock. Release by interruputs. (CPU restarts).
- SLEEP0 mode:

  CPU stops, and only the Time-Based-Timer(TBT) on peripherals operate using low frequency clock. Released when the reference time set to TBT has elapsed.
- SLEEP1 mode:

  CPU stops, and peripherals operate using low frequency clock. Release by interruput.(CPU restarts).

26. Wide operation voltage:

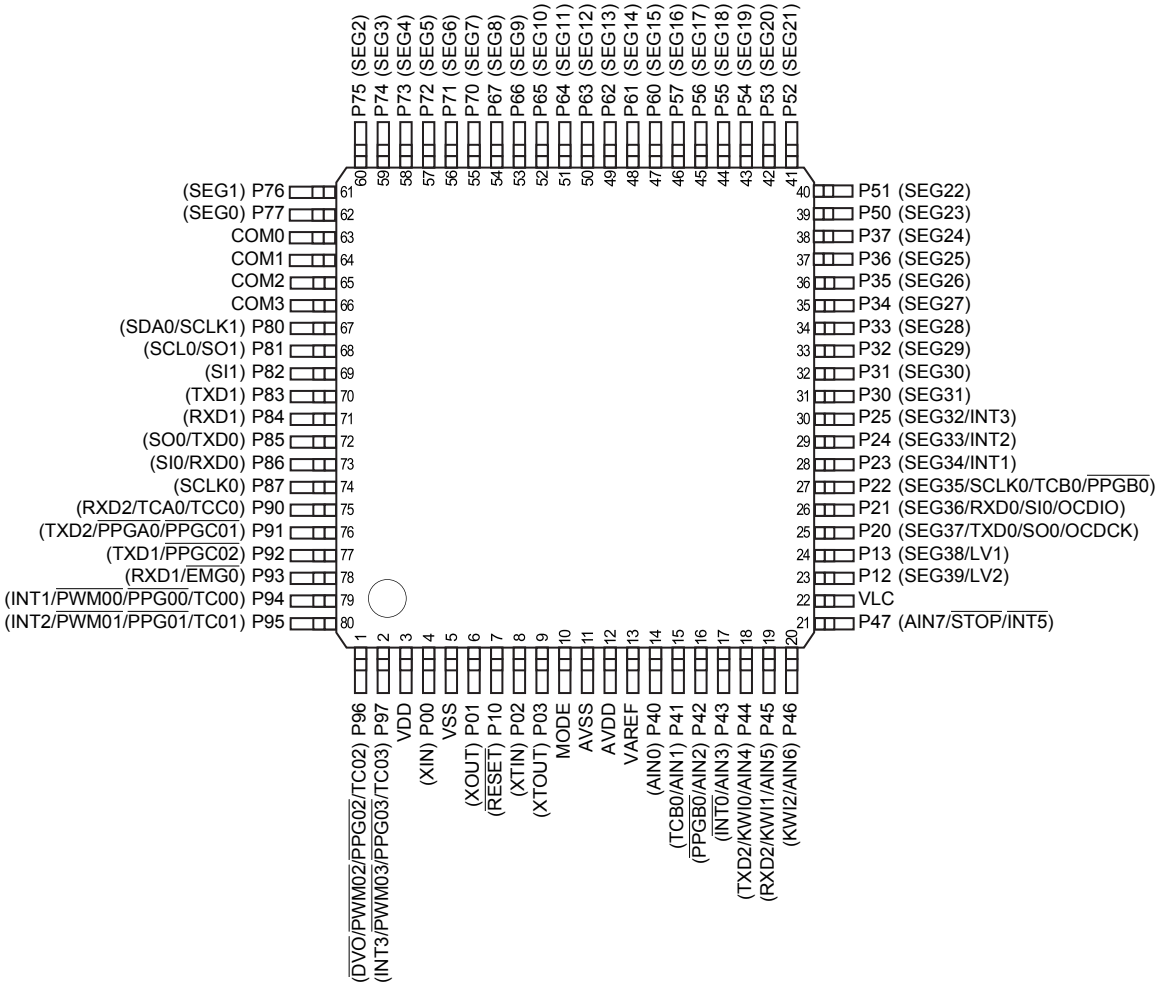
2.7 V to 5.5 V at 16MHz /32.768 kHz

1.8 V to 5.5 V at 8 MHz /32.768 kHz

Downloaded from Arrow.com.

1.2 Pin Assignment

TMP89FW24AFG (LQFP80-P-1212-0.50F)



Downloaded from Arrow.com.

TMP89FW24ADFG (QFP80-P-1420-0.80M)

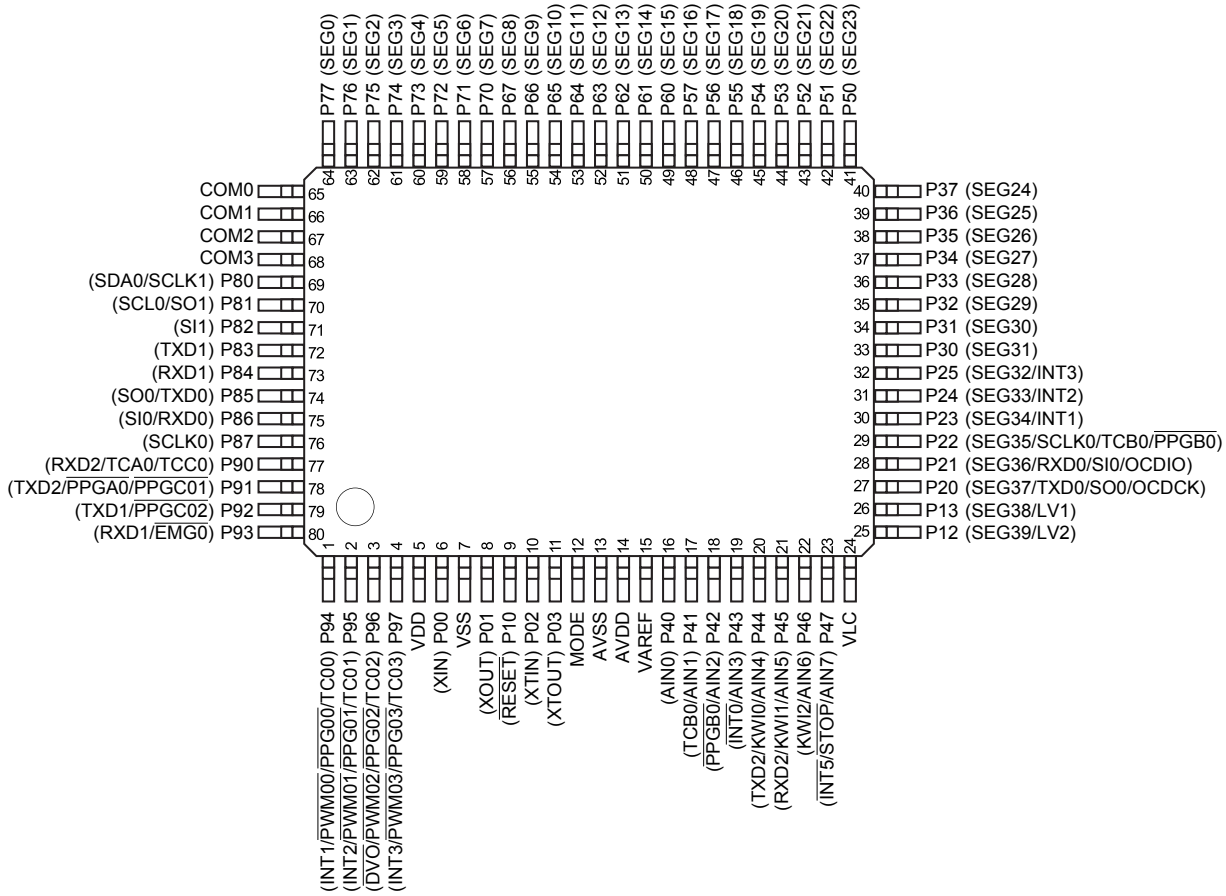


Figure 1-1 Pin Assignment

Downloaded from Arrow.com.

1.3 Block Diagram

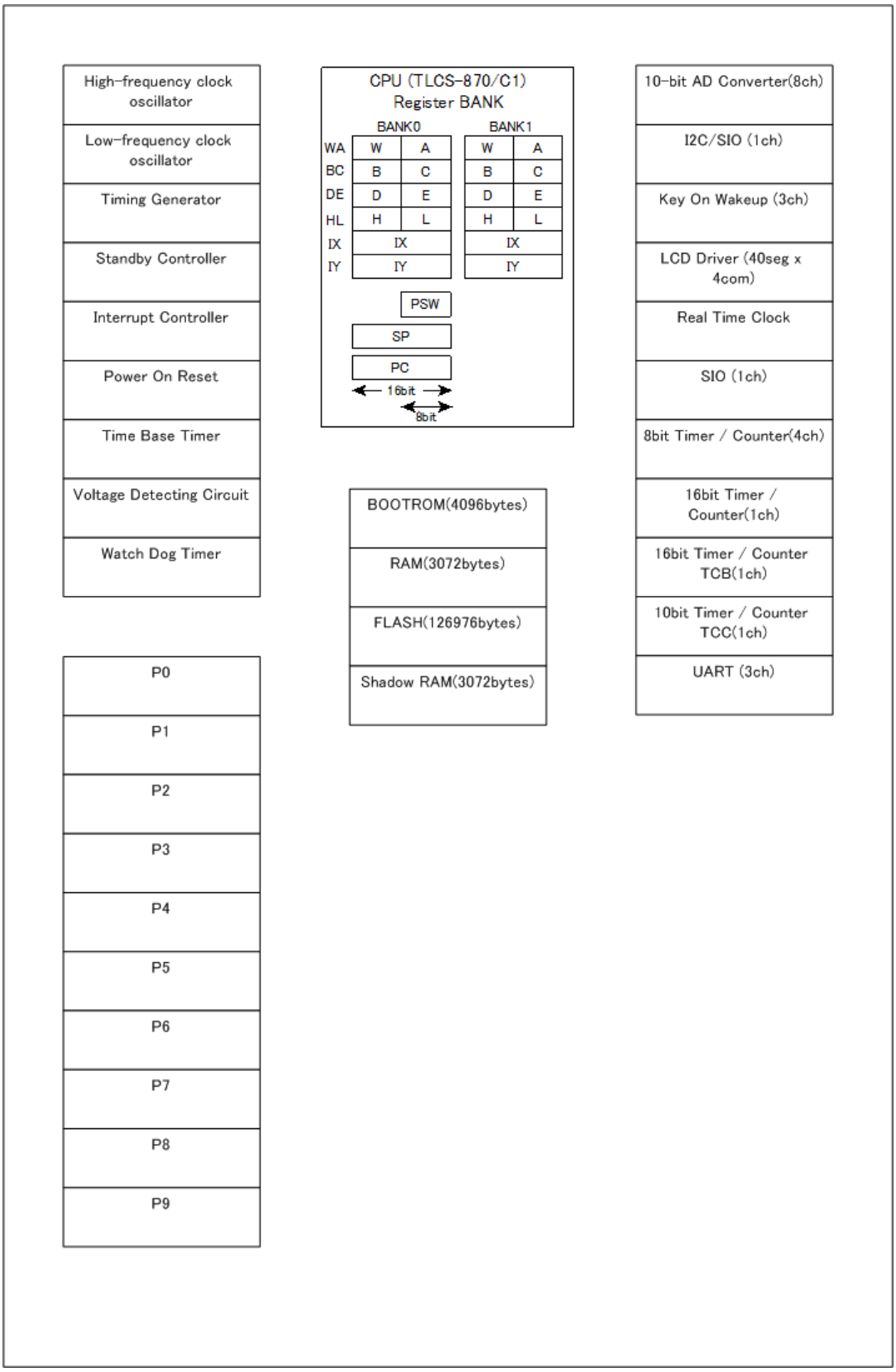


Figure 1-2 Block Diagram

Downloaded from Arrow.com.

1.4 Pin Names and Functions

The TMP89FW24A has MCU mode, parallel PROM mode, and serial PROM mode. Table 1-1 shows the pin functions in MCU mode. The serial PROM mode is explained later in a separate chapter.

Table 1-1 Pin Names and Functions (1/5)

Pin Name	Input/Output	Functions
P03 XTOUT	IO O	PORT03 Low frequency OSC output
P02 XTIN	IO I	PORT02 Low frequency OSC input
P01 XOUT	IO O	PORT01 High frequency OSC output
P00 XIN	IO I	PORT00 High frequency OSC input
P13 SEG38 LV1	IO O I	PORT13 LCD segment output 38 External bleeder resistance connection pins.
P12 SEG39 LV2	IO O I	PORT12 LCD segment output 39 External bleeder resistance connection pins.
P10 $\overline{\text{RESET}}$	IO I	PORT10 Reset signal input
P25 SEG32 INT3	IO O I	PORT25 LCD segment output 32 External interrupt 3 input
P24 SEG33 INT2	IO O I	PORT24 LCD segment output 33 External interrupt 2 input
P23 SEG34 INT1	IO O I	PORT23 LCD segment output 34 External interrupt 1 input
P22 SEG35 SCLK0 TCB0 $\overline{\text{PPGB0}}$	IO O IO I O	PORT22 LCD segment output 35 Serial clock input/output 0 TCB0 input PPGB0 output
P21 SEG36 RXD0 SI0 OCDIO	IO O I I IO	PORT21 LCD segment output 36 UART data input 0 Serial data input 0 OCD data input/output
P20 SEG37 TXD0 SO0 OCDCK	IO O O O I	PORT20 LCD segment output 37 UART data output 0 Serial data output 0 OCD clock input

Downloaded from Arrow.com.

Table 1-2 Pin Names and Functions (2/5)

Pin Name	Input/Output	Functions
P37 SEG24	IO O	PORT37 LCD segment output 24
P36 SEG25	IO O	PORT36 LCD segment output 25
P35 SEG26	IO O	PORT35 LCD segment output 26
P34 SEG27	IO O	PORT34 LCD segment output 27
P33 SEG28	IO O	PORT33 LCD segment output 28
P32 SEG29	IO O	PORT32 LCD segment output 29
P31 SEG30	IO O	PORT31 LCD segment output 30
P30 SEG31	IO O	PORT30 LCD segment output 31
P47 AIN7 $\overline{\text{STOP}}$ $\overline{\text{INT5}}$	IO I I I	PORT47 Analog input 7 STOP mode release input External interrupt 5 input
P46 AIN6 KWI2	IO I I	PORT46 Analog input 6 Key-on wake-up input 2
P45 AIN5 KWI1 RXD2	IO I I I	PORT45 Analog input 5 Key-on wake-up input 1 UART data input 2
P44 AIN4 KWI0 TXD2	IO I I O	PORT44 Analog input 4 Key-on wake-up input 0 UART data output 2
P43 AIN3 $\overline{\text{INT0}}$	IO I I	PORT43 Analog input 3 External interrupt 0 input
P42 AIN2 $\overline{\text{PPGB0}}$	IO I O	PORT42 Analog input 2 PPGB0 output
P41 AIN1 TCB0	IO I I	PORT41 Analog input 1 TCB0 input
P40 AIN0	IO I	PORT40 Analog input 0

Downloaded from Arrow.com.

Table 1-2 Pin Names and Functions (3/5)

Pin Name	Input/Output	Functions
P57 SEG16	IO O	PORT57 LCD segment output 16
P56 SEG17	IO O	PORT56 LCD segment output 17
P55 SEG18	IO O	PORT55 LCD segment output 18
P54 SEG19	IO O	PORT54 LCD segment output 19
P53 SEG20	IO O	PORT53 LCD segment output 20
P52 SEG21	IO O	PORT52 LCD segment output 21
P51 SEG22	IO O	PORT51 LCD segment output 22
P50 SEG23	IO O	PORT50 LCD segment output 23
P67 SEG8	IO O	PORT67 LCD segment output 8
P66 SEG9	IO O	PORT66 LCD segment output 9
P65 SEG10	IO O	PORT65 LCD segment output 10
P64 SEG11	IO O	PORT64 LCD segment output 11
P63 SEG12	IO O	PORT63 LCD segment output 12
P62 SEG13	IO O	PORT62 LCD segment output 13
P61 SEG14	IO O	PORT61 LCD segment output 14
P60 SEG15	IO O	PORT60 LCD segment output 15
P77 SEG0	IO O	PORT77 LCD segment output 0
P76 SEG1	IO O	PORT76 LCD segment output 1
P75 SEG2	IO O	PORT75 LCD segment output 2
P74 SEG3	IO O	PORT74 LCD segment output 3
P73 SEG4	IO O	PORT73 LCD segment output 4

Downloaded from Arrow.com.

Table 1-2 Pin Names and Functions (4/5)

Pin Name	Input/Output	Functions
P72 SEG5	IO O	PORT72 LCD segment output 5
P71 SEG6	IO O	PORT71 LCD segment output 6
P70 SEG7	IO O	PORT70 LCD segment output 7
P87 SCLK0	IO IO	PORT87 Serial clock input/output 0
P86 RXD0 SI0	IO I I	PORT86 UART data input 0 Serial data input 0
P85 TXD0 SO0	IO O O	PORT85 UART data output 0 Serial data output 0
P84 RXD1	IO I	PORT84 UART data input 1
P83 TXD1	IO O	PORT83 UART data output 1
P82 SI1	IO I	PORT82 Serial data input 1
P81 SO1 SCL0	IO O IO	PORT81 Serial data output 1 I2C bus clock input/output 0
P80 SCLK1 SDA0	IO IO IO	PORT80 Serial clock input/output 1 I2C bus data input/output 0
P97 TC03 $\overline{\text{PPG03}}$ $\overline{\text{PWM03}}$ INT3	IO I O O I	PORT97 TC03 input PPG03 output PWM03 output External interrupt 3 input
P96 TC02 $\overline{\text{PPG02}}$ $\overline{\text{PWM02}}$ $\overline{\text{DVO}}$	IO I O O O	PORT96 TC02 input PPG02 output PWM02 output Divider output
P95 TC01 $\overline{\text{PPG01}}$ $\overline{\text{PWM01}}$ INT2	IO I O O I	PORT95 TC01 input PPG01 output PWM01 output External interrupt 2 input

Downloaded from Arrow.com.

Table 1-2 Pin Names and Functions (5/5)

Pin Name	Input/Output	Functions
P94 TC00 $\overline{\text{PPG00}}$ $\overline{\text{PWM00}}$ INT1	IO I O O I	PORT94 TC00 input PPG00 output PWM00 output External interrupt 1 input
P93 $\overline{\text{EMG0}}$ RXD1	I I I	PORT93 Emergency stop input0 UART data input 1
P92 $\overline{\text{PPGC02}}$ TXD1	IO O O	PORT92 PPGC02 output UART data output 1
P91 $\overline{\text{PPGC01}}$ $\overline{\text{PPGA0}}$ TXD2	IO O O O	PORT91 PPGC01 output PPGA0 output UART data output 2
P90 TCC0 TCA0 RXD2	IO I I I	PORT90 TCC0 input TCA0 input UART data input 2
COM3	O	LCD common output 3
COM2	O	LCD common output 2
COM1	O	LCD common output 1
COM0	O	LCD common output 0
MODE	I	Test pin for out-going test (fix to Low level).
VAREF	I	Analog reference voltage input pin for A/D conversion.
AVDD	I	Analog power supply pin.
VLC	I	Power supply pin for LCD driver.
AVSS	I	Analog GND pin
VDD	I	VDD pin
VSS	I	GND pin

Downloaded from Arrow.com.

Downloaded from Arrow.com.

2. CPU Core

2.1 Configuration

The CPU core consists of a CPU, a system clock controller and a reset circuit.

This chapter describes the CPU core address space, the system clock controller and the reset circuit.

2.2 Memory space

The 870/C1 CPU memory space consists of a code area to be accessed as instruction operation codes and operands and a data area to be accessed as sources and destinations of transfer and calculation instructions.

Both the code and data areas have independent 64-Kbyte address spaces.

2.2.1 Code area

The code area stores operation codes, operands, vector tables for vector call instructions and interrupt vector tables.

The RAM, the BOOTROM and the Flash are mapped in the code area.

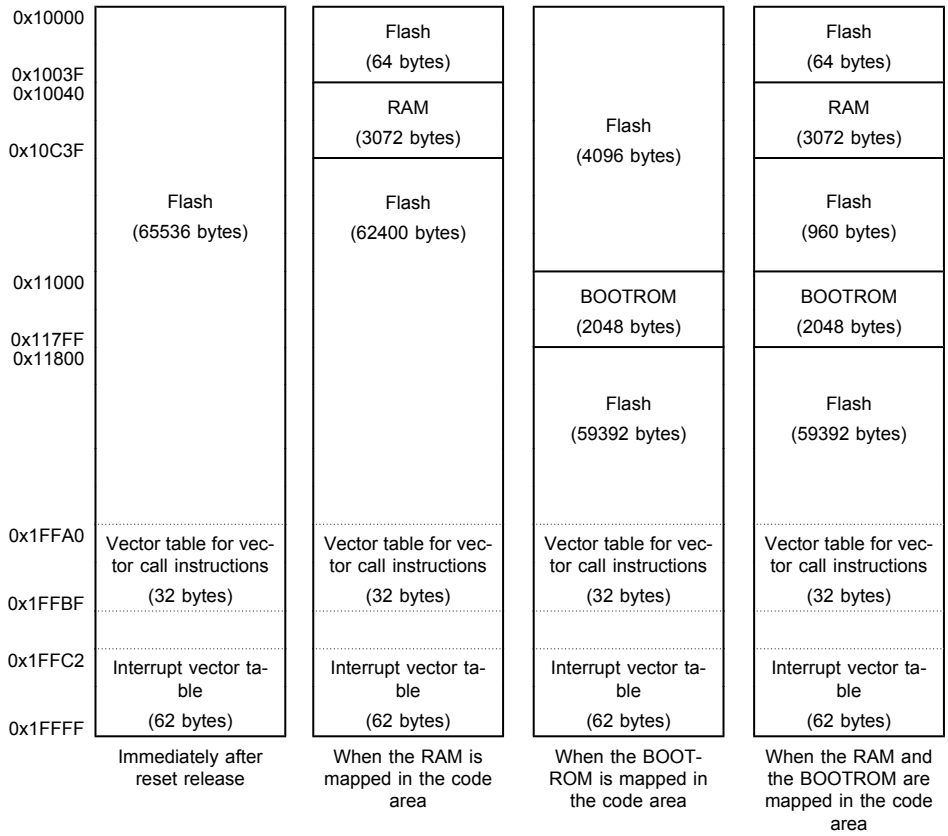


Figure 2-1 Memory Map in the Code Area

Note: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

Downloaded from Arrow.com.

2.2.1.1 RAM

The RAM is mapped in the data area immediately after reset release.

By setting SYSCR3<RAREA> to "1" and writing 0xD4 to SYSCR4, RAM can be mapped to 0x10040to 0x10C3F in the code area to execute the program.

At this time, by setting SYSCR<RVCTR> to "1" and writing 0xD4 to SYSCR4, vector table for vector call instructions and interrupt except reset can be mapped to RAM.

In the serial PROM mode, the RAM is mapped to 0x10040 to 0x10C3F in the code area, regardless of the value of SYSCR3<RAREA>. The program can be executed on the RAM using the RAM loader function.

Note1: The contents of the RAM become unstable when the power is turned on and immediately after a reset is released. To execute the program by using the RAM, transfer the program to be executed in the initialization routine.

System control register 3

SYSCR3 (0x00FDE)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	RVCTR	RAREA	(RSTDIS)
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

RAREA	Specifies mapping of the RAM in the code area	0 :	The RAM is not mapped from 0x10040 to 0x10C3F in the code area.	
		1 :	The RAM is mapped from 0x10040 to 0x10C3F in the code area.	
RVCTR	Specifies mapping of the vector table for vector call instructions and interrupts		Vector table for vector call instructions	Vector table for interrupt
		0 :	0x1FFA0 to 0x1FFBF in the code area	0x1FFC2 to 0x1FFFF in the code area
		1 :	0x101A0 to 0x101BF in the code area	0x101C2 to 0x101FD in the code area

Note 1: The value of SYSCR3<RAREA> is invalid until 0xD4 is written into SYSCR4.

Note 2: To assign vector address areas to RAM, set SYSCR3<RVCTR> to "1" and SYSCR3<RAREA> to "1".

Note 3: Do not set SYSCR3<RVCTR> to "0" by using the RAM loader program. If an interrupt occurs with SYSCR3<RVCTR> set to "0", the BOOTROM area is referenced as a vector address and, therefore, the program will not function properly.

Note 4: Bits 7 to 3 of SYSCR3 are read as "0".

System control register 4

SYSCR4 (0x00FDF)

	7	6	5	4	3	2	1	0
Bit Symbol	SYSCR4							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

SYSCR4	Writes the SYSCR3 data control code.	0xB2 :	Enables the contents of SYSCR3<RSTDIS>.
		0xD4 :	Enables the contents of SYSCR3<RAREA> and SYSCR3 <RVCTR>.
		0x71 :	Enables the contents of IRSTSR<FCLR>
			Others : Invalid

Note 1: SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 2: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Downloaded from Arrow.com.

Note 3: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

System control status register 4

SYSSR4 (0x00FDF)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	RVCTRS	RAREAS	(RSTDIS)
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

RAREAS	Status of mapping of the RAM in the code area	0 : 1 :	The enabled SYSCR3<RAREA> data is "0". The enabled SYSCR3<RAREA> data is "1".
RVCTRS	Status of mapping of the vector address in the area	0 : 1 :	The enabled SYSCR3<RVCTR> data is "0". The enabled SYSCR3<RVCTR> data is "1".

Note: Bits 7 to 3 of SYSSR4 are read as "0".

Example: Program transfer (Transfer the program saved in the data area to the RAM.)

```
            LD    HL, TRANSFER_START_ADDRESS    ;Destination RAM address
            LD    DE, PROGRAM_START_ADDRESS     ;Source ROM address
            LD    BC, BYTE_OF_PROGRAM           ;Number of bytes of the program to be executed -1
TRANS_RAM:  LD    A, (DE)                       ;Reading the program to be transferred
            LD    (HL), A                       ;Writing the program to be transferred
            INC   HL                            ;Destination address increment
            INC   DE                            ;Source address increment
            DEC   BC                            ;Have all the programs been transferred?
            J     F, code_addr(TRANS_RAM)
```

2.2.1.2 BOOTROM

The BOOTROM is not mapped in the code area or the data area after reset release.

Setting FLSCR1<BAREA> to "1" and writing 0xD5 to FLSCR2 maps the BOOTROM to 0x11000 to 0x117FF in the code area and to 0x01000 to 0x017FF in the data area. Flash memory can be easily programmed by using the API (Application Programming Interface) contained in the BOOTROM.

Note 1: When the BOOTROM is not mapped in the code area, an instruction is fetched from the Flash or an SWI instruction is fetched, depending on the capacity of the internal Flash.

Note 2: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

Flash memory control register 1

FLSCR1 (0x00FD0)

	7	6	5	4	3	2	1	0
Bit Symbol	(FLSMD)			BAREA	(FAREA)		(ROMSEL)	
Read/Write	R/W			R/W	R/W		R/W	
After reset	0	1	0	0	0	0	0	0

BAREA	Specifies mapping of the BOOTROM in the code and data areas	0 : 1 :	The BOOTROM is not mapped to 0x11000 to 0x117FF in the code area and to 0x01000 to 0x017FF in the data area. The BOOTROM is mapped to 0x11000 to 0x117FF in the code area and to 0x01000 to 0x017FF in the data area.

Downloaded from Arrow.com.

Note: The flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting to be reflected and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2. The value of the shift register can be checked by reading the register FLSCRM.

Flash memory control register 2

FLSCR2		7	6	5	4	3	2	1	0
(0x00FD1)	Bit Symbol	CR1EN							
	Read/Write	W							
	After reset	*	*	*	*	*	*	*	*

CR1EN	FLSCR1 register enable/disable control	0xD5	Enable a change in the FLSCR1 setting
		Others	Reserved

2.2.1.3 Flash

The Flash is mapped to 0x10000 to 0x1FFFF in the code area after reset release.

Downloaded from Arrow.com.

2.2.2 Data area

The data area stores the data to be accessed as sources and destinations of transfer and calculation instructions.

The SFR, the RAM, the BOOTROM and the FLASH are mapped in the data area.

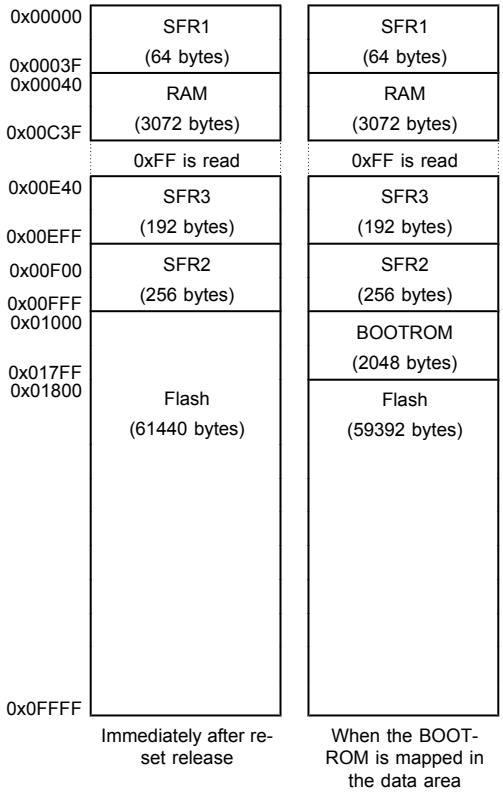


Figure 2-2 Memory Map in the Data Area

Note: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

2.2.2.1 SFR

The SFR is mapped to 0x00000 to 0x0003F (SFR1), 0x00F00 to 0x00FFF (SFR2) and 0x00E40 to 0x00EFF (SFR3) in the data area after reset release.

Note: Don't access the reserved SFR.

2.2.2.2 RAM

The RAM is mapped to 0x00040 to 0x00C3F in the data area after reset release.

Note: The contents of the RAM become unstable when the power is turned on and immediately after a reset is released. To execute the program by using the RAM, transfer the program to be executed in the initialization routine.

Downloaded from Arrow.com.

Example: RAM initialization program

```
        LD      HL, RAM_TOP_ADDRESS         ;Head of address of the RAM to be initialized
        LD      A, 0x00                     ;Initialization data
        LD      BC, BYTE_OF_CLEAR_BYTES     ;Number of bytes of RAM to be initialized -1
CLR_RAM: LD     (HL), A                     ;Initialization of the RAM
        INC     HL                          ;Initialization address increment
        DEC     BC                          ;Have all the RAMs been initialized?
        J       F, code_addr(CLR_RAM)
```

2.2.2.3 BOOTROM

The BOOTROM is not mapped in the code area or the data area after reset release.

Setting FLSCR1<BAREA> to "1" and writing 0xD5 to FLSCR2 maps the BOOTROM to 0x01000 to 0x017FF in the code area and to 0x01000 to 0x017FF in the data area. Flash memory can be easily programmed by using the API (Application Programming Interface) contained in the BOOTROM.

Note1: Only the first 2 Kbytes of the BOOTROM are mapped in the memory map, except in the serial PROM mode.

Flash memory control register 1

FLSCR1 (0x00FD0)

	7	6	5	4	3	2	1	0
Bit Symbol	(FLSMD)			BAREA	(FAREA)		(ROMSEL)	
Read/Write	R/W			R/W	R/W		R/W	
After reset	0	1	0	0	0	0	0	0

BAREA	Specifies mapping of the BOOT-ROM in the code and data areas	0 :	The BOOTROM is not mapped to 0x11000 to 0x117FF in the code area and to 0x01000 to 0x017FF in the data area.
		1 :	The BOOTROM is mapped to 0x11000 to 0x117FF in the code area and to 0x01000 to 0x017FF in the data area.

Note: The flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting to be reflected and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2. The value of the shift register can be checked by reading the register FLSCRM.

Flash memory control register 2

FLSCR2 (0x00FD1)

	7	6	5	4	3	2	1	0
Bit Symbol	CR1EN							
Read/Write	W							
After reset	*	*	*	*	*	*	*	*

CR1EN	FLSCR1 register enable/disable control	0xD5	Enable a change in the FLSCR1 setting
		Others	Reserved

2.2.2.4 Flash

The Flash is mapped to 0x01000 to 0x0FFFF in the data area after reset release.

Downloaded from Arrow.com.

2.3 System clock controller

2.3.1 Configuration

The system clock controller consists of a clock generator, a clock gear, a timing generator, a warm-up counter and an operation mode control circuit.

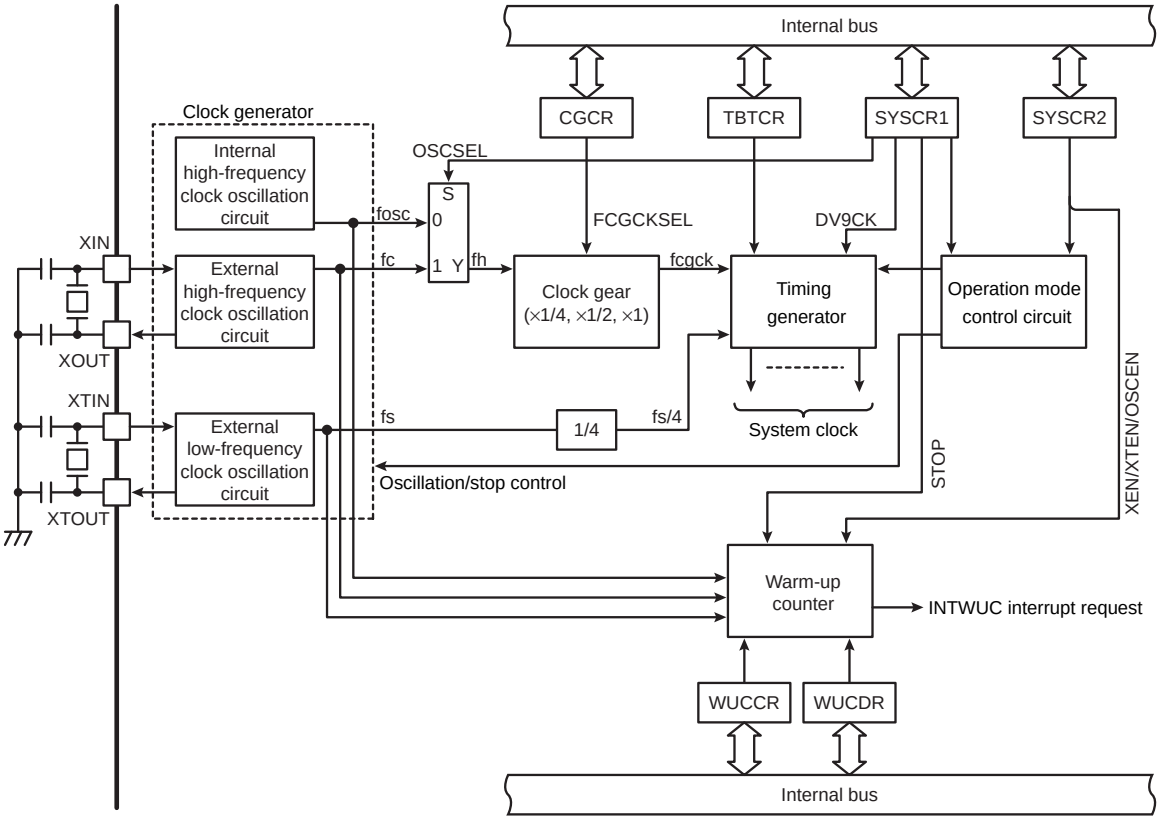


Figure 2-3 System Clock Controller

2.3.2 Control

The system clock controller is controlled by system control register 1 (SYSCR1), system control register 2 (SYSCR2), the warm-up counter control register (WUCCR), the warm-up counter data register (WUCDR) and the clock gear control register (CGCR).

Downloaded from Arrow.com.

System control register 1

SYSCR1 (0x00FDC)

	7	6	5	4	3	2	1	0
Bit Symbol	STOP	RELM	OUTEN	DV9CK	OSCSEL	-	-	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R
After reset	0	0	0	0	0	0	0	0

STOP	Activates the STOP mode	0 : 1 :	Operate the CPU and the peripheral circuits Stop the CPU and the peripheral circuits (activate the STOP mode)
RELM	Selects the STOP mode release method	0 : 1 :	Edge-sensitive release mode (Release the STOP mode at the rising edge of the STOP mode release signal) Level-sensitive release mode (Release the STOP mode at the "H" level of the STOP mode release signal)
OUTEN	Selects the port output state in the STOP mode	0 : 1 :	High impedance Output hold
DV9CK	Selects the input clock to stage 9 of the divider	0 : 1 :	fcgck/2^9 fs/4
OSCSEL	Selects the high-frequency reference clock (fh)	0 : 1 :	Internal high-frequency clock (fosc) External high-frequency clock (fc)

Note 1: fosc: Internal high-frequency clock [Hz], fc: External high-frequency clock [Hz], fcgck: Gear clock [Hz], fs: External low-frequency clock [Hz]

Note 2: Bits 2, 1 and 0 of SYSCR1 are read as "0".

Note 3: If the STOP mode is activated with SYSCR1<OUTEN> set at "0", the port internal input is fixed to "0". Therefore, an external interrupt may be set at the falling edge, depending on the pin state when the STOP mode is activated.

Note 4: The P47 pin is also used as the STOP pin. When the STOP mode is activated, the pin reverts to high impedance state and is put in input mode, regardless of the state of SYSCR1<OUTEN>.

Note 5: Writing of the second byte data will be executed improperly if the operation is switched to the STOP state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 6: Don't set SYSCK1<DV9CK> to "1" before the oscillation of the external low-frequency clock oscillation circuit becomes stable.

Note 7: In the SLOW1/2 or SLEEP1 mode, fs/4 is input to stage 9 of the divider, regardless of the state of SYSCR1< DV9CK >.

Note 8: SYSCR1<OSCSEL> should be set while SYSCR2<SYSCK> is "0" (during the NORMAL1 or NORMAL2 mode). Writing to SYSCR1<OSCSEL> while SYSCR2<SYSCK> = "1" (during the SLOW1 or SLOW2 mode) has no effect.

System control register 2

SYSCR2 (0x00FDD)

	7	6	5	4	3	2	1	0
Bit Symbol	OSCEN	XEN	XTEN	SYSCK	IDLE	TGHALT	-	-
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R
After reset	1	0	0	0	0	0	0	0

OSCEN	Controls the internal high-frequency clock (fosc)	0 : 1 :	Stop oscillation Continue or start oscillation
XEN	Controls the external high-frequency clock (fc)	0 : 1 :	Stop oscillation Continue or start oscillation
XTEN	Controls the external low-frequency clock (fs)	0 : 1 :	Stop oscillation Continue or start oscillation
SYSCK	Selects a system clock	0 : 1 :	Gear clock (fcgck) (NORMAL1/2 or IDLE1/2 mode) External low-frequency clock (fs/4) (SLOW1/2 or SLEEP1 mode)
IDLE	CPU and WDT control (IDLE1/2 or SLEEP1 mode)	0 : 1 :	Operate the CPU and the WDT Stop the CPU and the WDT (Activate IDLE1/2 or SLEEP1 mode)
TGHALT	TG control (IDLE0 or SLEEP0 mode)	0 : 1 :	Enable the clock supply from the TG to all the peripheral circuits Disable the clock supply from the TG to the peripheral circuits except the TBT (Activate IDLE0 or SLEEP0 mode)

Downloaded from Arrow.com.

Note 1: fosc: Internal high-frequency clock [Hz], fc: External high-frequency clock [Hz], fcgck: Gear clock [Hz], fs: External low-frequency clock [Hz]

Note 2: WDT: Watchdog timer, TG: Timing generator

Note 3: Don't set both SYSCR2<IDLE> and SYSCR2<TGHALT> to "1" simultaneously.

Note 4: Writing of the second byte data will be executed improperly if the operation is switched to the IDLE state by an instruction, such as LDW, which executes 2-byte data transfer at a time.

Note 5: When the IDLE1/2 or SLEEP1 mode is released, SYSCR2<IDLE> is cleared to "0" automatically.

Note 6: When the IDLE0 or SLEEP0 mode is released, SYSCR2<TGHALT> is cleared to "0" automatically.

Note 7: Bits 1 and 0 of SYSCR2 are read as "0".

Note 8: Do not set both SYSCR2<OSCEN> and SYSCR2<XEN> to "1" simultaneously except when switching the high-frequency reference clock (fh). (When the switching of the reference clock (fh) is complete, one of the two high-frequency clocks not to be used should be stopped.)

Warm-up counter control register

WUCCR (0x00FCD)

	7	6	5	4	3	2	1	0
Bit Symbol	WUCRST	-	-	-	WUCDIV		WUCSEL	
Read/Write	W	R	R	R	R/W		R/W	
After reset	0	0	0	0	1	1	0	0

WUCRST	Resets and stops the warm-up counter	0 : 1 :	- Clear and stop the counter
WUCDIV	Selects the frequency division of the warm-up counter source clock	00 : 01 : 10 : 11 :	Source clock Source clock / 2 Source clock / 2^2 Source clock / 2^3
WUCSEL	Selects the warm-up counter source clock	00 : 01 : 10 : 11 :	Select the internal high-frequency clock (fosc) Select the external high-frequency clock (fc) Select the external low-frequency clock (fs) Reserved

Note 1: fosc : Internal high-frequency clock [Hz], fc: External high-frequency clock [Hz], fcgck: Gear clock [Hz], fs: External low-frequency clock [Hz]

Note 2: WUCCR<WUCRST> is cleared to "0" automatically, and need not be cleared to "0" after being set to "1".

Note 3: Bits 7 to 4 of WUCCR are read as "0".

Note 4: Before starting the warm-up counter operation, set the source clock and the frequency division rate at WUCCR<WUCSEL, WUCDIV> and set the warm-up time at WUCDR.

Warm-up counter data register

WUCDR (0x00FCE)

	7	6	5	4	3	2	1	0
Bit Symbol	WUCDR							
Read/Write	R/W							
After reset	0	1	1	0	0	1	1	0

WUCDR	Warm-up time setting

Note 1: Don't start the warm-up counter operation with WUCDR set at "0x00".

Downloaded from Arrow.com.

Clock gear control register

CGCR (0x00FCF)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	-	FCGCKSEL	
Read/Write	R	R	R	R	R	R	R/W	
After reset	0	0	0	0	0	0	0	0

FCGCKSEL	Clock gear setting	00 : 01 : 10 : 11 :	fcgck = fh / 4 fcgck = fh / 2 fcgck = fh Reserved

Note 1: fh: High-frequency reference clock [Hz], fcgck: Gear clock [Hz]
Note 2: Don't change CGCR<FCGCKSEL> in the SLOW mode.
Note 3: Bits 7 to 2 of CGCR are read as "0".

2.3.3 Functions

2.3.3.1 Clock generator

The clock generator generates the basic clock for the system clocks to be supplied to the CPU core and peripheral circuits.

It contains three oscillation circuits: one for the internal high-frequency clock, one for the external high-frequency clock and one for the external low-frequency clock.

The oscillation circuit pins are also used as ports P0. For the setting to use them as ports, refer to the chapter of I/O Ports.

To use ports P00 and P01 for the external high-frequency clock oscillation circuit (as the XIN and XOUT pins), set P0FC0 to "1" and then set SYSCR2<XEN> to "1".

To use ports P02 and P03 for the external low-frequency clock oscillation circuit (as the XTIN and XTOUT pins), set P0FC2 to "1" and then set SYSCR2<XTEN> to "1".

The external high-frequency (fc) clock and the external low-frequency (fs) clock can easily be obtained by connecting an oscillator between the XIN and XOUT pins and between the XTIN and XTOUT pins, respectively.

Enabling/disabling the oscillation of the external high-frequency clock oscillation circuit and the external low-frequency clock oscillation circuit and switching the pin function to ports are controlled by the software and hardware.

The software control is executed by SYSCR2<XEN>, SYSCR2<XTEN> and the P0 port function control register P0FC.

The hardware control is executed by reset release and the operation mode control circuit when the operation is switched to the STOP mode as described in "2.3.5 Operation mode control circuit".

Note: No hardware function is available for external direct monitoring of the basic clock. The oscillation frequency can be adjusted by programming the system to output pulses at a certain frequency to a port (for example, a clock output) with interrupts disabled and the watchdog timer disabled and monitoring the output. An adjustment program must be created in advance for a system that requires adjustment of the oscillation frequency.

To prevent the dead lock of the CPU core due to the software-controlled enabling/disabling of the oscillation, an internal factor reset is generated depending on the combination of values of the clock selected as the main system clock, SYSCR2<OSCEN, XEN, XTEN> and the P0 port function control register P0FC0.

Downloaded from Arrow.com.

Table 2-1 Prohibited Combinations of Oscillation Enable Register Conditions

P0FC0	SYSCR2 <OSCEN>	SYSCR2 <XEN>	SYSCR1 <OSCSEL>	SYSCR2 <XTEN>	SYSCR2 <SYSCK>	State
Don't Care	0	0	Don't Care	0	Don't Care	All the oscillation circuits are stopped.
Don't Care	Don't Care	Don't Care	Don't Care	0	1	The external low-frequency clock (fs) is selected as the main system clock, but the external low-frequency clock oscillation circuit is stopped.
Don't Care	0	Don't Care	0	Don't Care	0	The high-frequency reference clock (fh) is selected as the main system clock, but the internal high-frequency clock oscillation circuit (fosc) selected as fh is stopped.
Don't Care	Don't Care	0	1	Don't Care	0	The high-frequency reference clock (fh) is selected as the main system clock, but the external high-frequency clock oscillation circuit (fc) selected as fh is stopped.
0	Don't Care	1	Don't Care	Don't Care	Don't Care	The external high-frequency clock oscillation circuit (fc) is enabled, but the relevant port pins are set as general-purpose ports.

Note: It takes a certain period of time after SYSCR1<OSCSEL> and SYSCR2<SYSCK> are changed before the main system clock is switched. If the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 2-1 and a system clock reset occurs. For details of clock switching, refer to "2.3.6 Operation Mode Control".

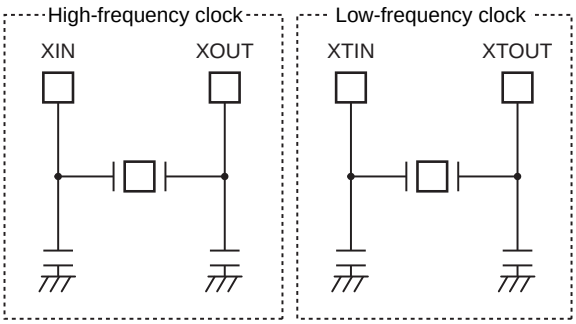


Figure 2-4 Examples of Oscillator Connection

(1) High-frequency reference clock (fh)

The high-frequency reference clock (fh) is used to operate the TMP89FW24A at high speed.

When SYSCR1<OSCSEL> = "1", the external high-frequency clock (fc) is used as the reference clock (fh). When SYSCR1<OSCSEL> = "0", the internal high-frequency clock (fosc) is used as the reference clock (fh). Upon reset release, SYSCR1<OSCSEL> is cleared to "0" and the internal high-frequency clock (fosc) is used as the reference clock (fh).

When the high-frequency reference clock (fh) is switched, both the external high-frequency clock (fc) and the internal high-frequency clock (fosc) need to be oscillating. To switch the high-frequency reference clock (fh), be sure to follow the steps explained below.

In the process of switching the high-frequency reference clock (fh), there is a time when both the external high-frequency clock (fc) and the internal high-frequency clock (fosc) are enabled simultaneously. Mode transitions, as explained in "1.3.5 Operation mode control circuit", must not be made in this state. Once the reference clock has been switched, be sure to stop either of the high-frequency clocks not to be used.

- Switching from fosc to fc

  With the relevant bits in the P0FC0 register set to "1", set SYSCR2<XEN> to "1" to enable the external high-frequency clock (fc).

Downloaded from Arrow.com.

After making sure that the external high-frequency clock (fc) has achieved stable oscillation by using the warm-up counter, set SYSCR1<OSCSEL> to "1".

A maximum of 2/fosc + 2.5/fc [s] after SYSCR1<OSCSEL> is set to "1", the high-frequency reference clock (fh) changes to the external high-frequency clock (fc).

After the reference clock (fh) has been switched, wait for at least 2 machine cycles, and then clear SYSCR2<OSCEN> to "0" to stop the internal high-frequency clock (fosc). If SYSCR2<OSCEN> is cleared to "0" while the reference clock (fh) is being switched, a system clock reset is generated.

Note 1: When the high-frequency reference clock (fh) is switched, the hardware synchronizes the external high-frequency clock (fc) and the internal high-frequency clock (fosc). While this is done, fh stops for a maximum of 2.5/fc [s].

Note 2: After changing SYSCR1<OSCSEL>, be sure to wait for at least 2 machine cycles before clearing SYSCR2<OSCEN> to "0". If SYSCR2<OSCEN> is cleared to "0" without waiting for at least 2 machine cycles, a system clock reset is generated.

Note 3: SYSCR1<OSCSEL> must be set while SYSCR2<SYSCK> = "0" (during the NORMAL1 or NORMAL2 mode). Writing to SYSCR1<OSCSEL> while SYSCR2<SYSCK> = "1" (during the SLOW1 or SLOW2 mode) has no effect.

Note 4: Setting SYSCR2<XEN> to "1" while P0FC0 = "0" generates a system clock reset.

Note 5: If SYSCR2<XEN> is set to "1" while SYSCR2<XEN> = "1", the warm-up counter does not start counting the source clock.

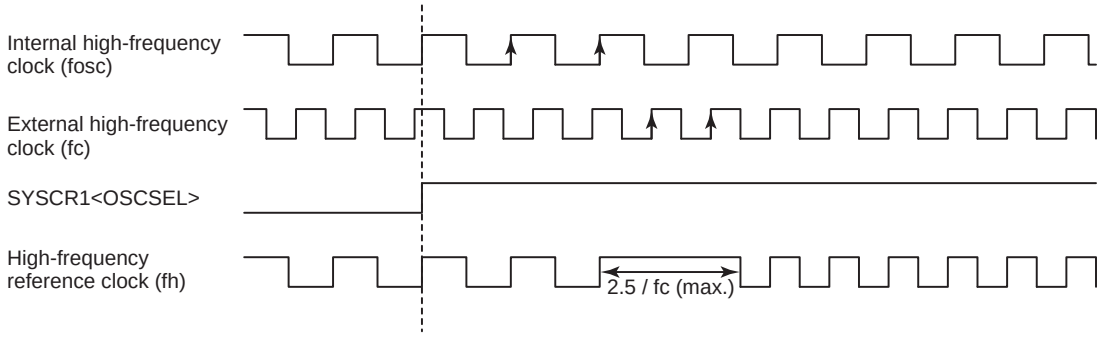


Figure 2-5 Switching the High-Frequency Reference Clock (fh) (fosc to fc)

Table 2-2 Steps for Switching the High-Frequency Reference Clock (fh) from fosc to fc

Step	P0FC0	SYSCR2 <OSCEN>	SYSCR2 <XEN>	SYSCR1 <OSCSEL>	Main system clock	State
1	0	1	0	0	fosc	The high-frequency reference clock is fosc, and ports P00 and P01 are used as I/O ports.
2	1	1	0	0	fosc	Ports P00 and P01 are set as oscillation pins.
3	1	1	1	0	fosc	The high-frequency clock oscillation circuit is warming up.
4	1	1	1	1	fosc→fc	The high-frequency reference clock is being switched to fc.
5	1	0	1	1	fc	The high-frequency reference clock has been switched to fc.

Note: Be sure to follow the above steps when switching the high-frequency reference clock.

Downloaded from Arrow.com.

Example: Setting ports P00 and P01 as oscillation pins and switching the high-frequency reference clock from fosc to fc (warm-up time: approx. 300 μs at fc = 8 MHz)

```
        LD      (WUCCR), 0y00000001     ;WUCCR<WUCDIV>←"00" (No division)
                                        ;WUCCR<WUCSEL>←"01" (Selects fc as the source clock)
        LD      (WUCDR), 0x26           ;Sets the warm-up time
                                        ;(Determine the time depending on the oscillator characteristics)
                                        ;300μs / 8 μs =37.5 round up to 0x26
        SET     (EIRL).4                ;Enables INTWUC interrupts
        SET     (P0FC).0                ;P0FC0←"1" (Set P00 and P01 as oscillation pins)
        SET     (SYSCR2). 6             ;SYSCR2<XTEN>←"1"
                                        ;(Starts the external high-frequency clock oscillation and starts the
        :       :                       ;warm-up counter)
PINTWUC:SET     (SYSCR1). 3             ;SYSCR1<OSCSEL>←"1"
                                        ;(Switches the high-frequency reference clock from fosc to fc)
        NOP                             ;Waits for 2 machine cycles
        NOP                             ;Waits for 2 machine cycles
        CLR     (SYSCR2). 7             ;SYSCR2<OSCEN>←"0" (Stops fosc)
        RETI
```


Downloaded from Arrow.com.

- Switching from fc to fosc

Set SYSCR1<OSCEN> to "1" to enable the internal high-frequency clock (fosc).

After making sure that the internal high-frequency clock (fosc) has achieved stable oscillation by using the warm-up counter, clear SYSCR1<OSCSEL> to "0".

A maximum of 2/fc + 2.5/fosc [s] after SYSCR1<OSCSEL> is cleared to "0", the high-frequency reference clock (fh) changes to the internal high-frequency clock (fosc).

After the reference clock (fh) has been switched, wait for at least 2 machine cycles, and then clear SYSCR2<XEN> to "0" to stop the external high-frequency clock (fc). If SYSCR2<XEN> is cleared to "0" while the reference clock (fh) is being switched, a system clock reset is generated.

Note 1: When the high-frequency reference clock (fh) is switched, the hardware synchronizes the external high-frequency clock (fc) and the internal high-frequency clock (fosc). While this is done, fh stops for a maximum of 2.5/fosc [s].

Note 2: After changing SYSCR1<OSCSEL>, be sure to wait for at least 2 machine cycles before clearing SYSCR2<XEN> to "0". If SYSCR2<XEN> is cleared to "0" without waiting for at least 2 machine cycles, a system clock reset is generated.

Note 3: SYSCR1<OSCSEL> must be set while SYSCR2<SYSCK> = "0" (during the NORMAL1 or NORMAL2 mode). Writing to SYSCR1<OSCSEL> while SYSCR2<SYSCK> = "1" (during the SLOW1 or SLOW2 mode) has no effect.

Note 4: Setting SYSCR2<XEN> to "1" while P0FC0 = "0" generates a system clock reset.

Note 5: If SYSCR2<XEN> is set to "1" while SYSCR2<XEN> = "1", the warm-up counter does not start counting the source clock.

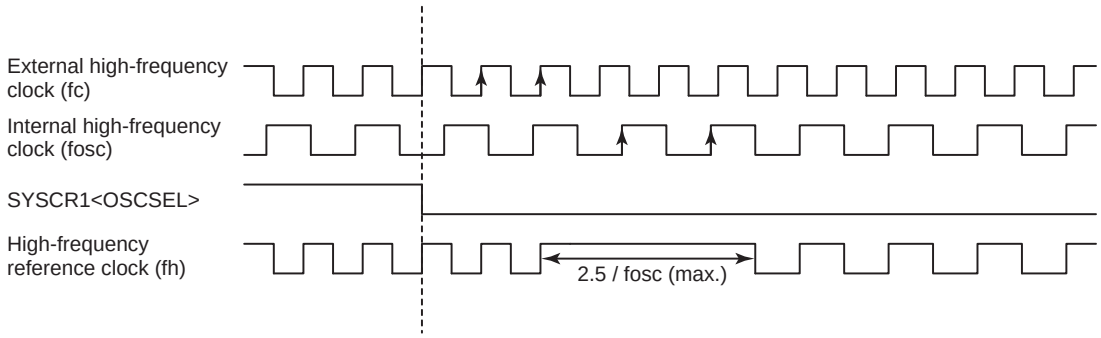


Figure 2-6 Switching the High-Frequency Reference Clock (fh) (fc to fosc)

Table 2-3 Steps for Switching the High-Frequency Reference Clock (fh) from fc to fosc

Step	P0FC0	SYSCR2 <OSCEN>	SYSCR2 <XEN>	SYSCR1 <OSCSEL>	Main system clock	State
1	1	0	1	1	fc	The high-frequency reference clock is fc.
2	1	1	1	1	fc	The high-frequency clock oscillation circuit is warming up.
3	1	1	1	0	fc→fosc	The high-frequency reference clock is being switched to fosc.
4	1	1	0	0	fosc	The high-frequency reference clock has been switched to fosc.

Note: Be sure to follow the above steps when switching the high-frequency reference clock.

2026-04-21
Downloaded from Arrow.com.

Example: Switching the high-frequency reference clock from fc to fosc (warm-up time: approx. 100 μs = at fosc = 5 MHz)

```
        LD      (WUCCR), 0y00000000     ;WUCCR<WUCDIV>←"00" (No division)
                                        ;WUCCR<WUCSEL>←"01" (Selects fosc as the source clock)
        LD      (WUCDR), 0x08           ;Sets the warm-up time
                                        ;(Determine the time depending on the oscillator characteristics)
                                        ;100μs / 12.8 μs = 7.8 round up to 0x08
        SET     (EIRL).4                ;Enables INTWUC interrupts
        SET     (SYSCR2). 7             ;SYSCR2<OSCEN>←"1"
                                        ;(Starts the internal high-frequency clock oscillation and starts the
        :       :                        warm-up counter)
PINTWUC:CLR     (SYSCR1). 3              SYSCR1<OSCSEL>←"0"
                                         (Switches the high-frequency reference clock from fc to fosc)
        NOP                              Waits for 2 machine cycles
        NOP                              Waits for 2 machine cycles
        CLR     (SYSCR2). 6              SYSCR2<XEN>←"0" (Stops fc)
        RETI
```

(2) Low-frequency reference clock (fs)

The low-frequency reference clock (fs) is used to operate the TMP89FW24A at low speed. Power consumption can be reduced.

2.3.3.2 Clock gear

The clock gear is a circuit that selects a gear clock (fcgck) obtained by dividing the high-frequency reference clock (fh) and inputs it to the timing generator.

Selects a divided clock at CGCR<FCGCKSEL>.

Two machine cycles are needed after CGCR<FCGCKSEL> is changed before the gear clock (fcgck) is changed.

The gear clock (fcgck) may be longer than the set clock width, immediately after CGCR<FCGCKSEL> is changed.

Immediately after reset release, the gear clock (fcgck) is set as fh/4, which is obtained by dividing the high-frequency reference clock (fh) by 4.

Table 2-4 Gear Clock (fcgck)

CGCR<FCGCKSEL>	fcgck
00	fh / 4
01	fh / 2
10	fh
11	Reserved

Note: Don't change CGCR<FCGCKSEL> in the SLOW mode. This may stop the gear clock (fcgck) from being changed.

Downloaded from Arrow.com.

2.3.3.3 Timing generator

The timing generator is a circuit that generates system clocks to be supplied to the CPU core and the peripheral circuits from the gear clock (fcgck) or the clock that is a quarter of the external low-frequency clock (fs). The timing generator has the following functions:

1. Generation of the main system clock (fm)
2. Generation of clocks for the timer counter, the time base timer and other peripheral circuits

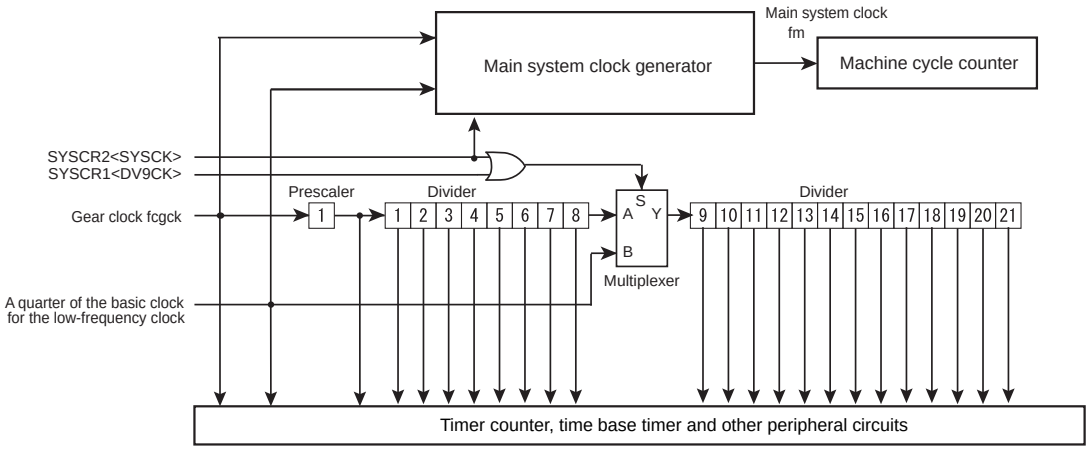


Figure 2-7 Configuration of Timing Generator

(1) Configuration of timing generator

The timing generator consists of a main system clock generator, a prescaler, a 21-stage divider and a machine cycle counter.

1. Main system clock generator

   This circuit selects the gear clock (fcgck) or the clock that is a quarter of the external low-frequency clock (fs) for the main system clock (fm) to operate the CPU core.

   Clearing SYSCR2<SYSCK> to "0" selects the gear clock (fcgck). Setting it to "1" selects the clock that is a quarter of the external low-frequency clock (fs).

   It takes a certain period of time after SYSCR2<SYSCK> is changed before the main system clock is switched. If the currently operating oscillation circuit is stopped before the main system clock is switched, the internal condition becomes as shown in Table 2-1 and a system clock reset occurs. For details of clock switching, refer to "2.3.6 Operation Mode Control".

2. Prescaler and divider

   These circuits divide fcgck. The divided clocks are supplied to the timer counter, the time base timer and other peripheral circuits.

   When both SYSCR1<DV9CK> and SYSCR2<SYSCK> are "0", the input clock to stage 9 of the divider becomes the output of stage 8 of the divider.

   When SYSCR1<DV9CK> or SYSCR2<SYSCK> is "1", the input clock to stage 9 of the divider becomes fs/4. When SYSCR2<SYSCK> is "1", the outputs of stages 1 to 8 of the divider and prescaler are stopped.

   The prescaler and divider are cleared to "0" at a reset and at the end of the warm-up operation that follows the release of STOP mode.

2026-04-21
Downloaded from Arrow.com.

3. Machine cycle

Instruction execution is synchronized with the main system clock (fm).

The minimum instruction execution unit is called a "machine cycle". One machine cycle corresponds to one main system clock.

There are a total of 11 different types of instructions for the TLCS-870/C1 Series: 10 types ranging from 1-cycle instructions, which require one machine cycle for execution, to 10-cycle instructions, which require 10 machine cycles for execution, and 13-cycle instructions, which require 13 machine cycles for execution.

2.3.4 Warm-up counter

The warm-up counter is a circuit that counts the internal high-frequency clock (fosc), the external high-frequency clock (fc) and the external low-frequency clock (fs), and it consists of a source clock selection circuit, a 3-stage frequency division circuit and a 14-stage counter.

The warm-up counter is used to secure the time after a power-on reset is released before the supply voltage becomes stable and secure the time after the STOP mode is released or the operation mode is changed before the oscillation by the oscillation circuit becomes stable.

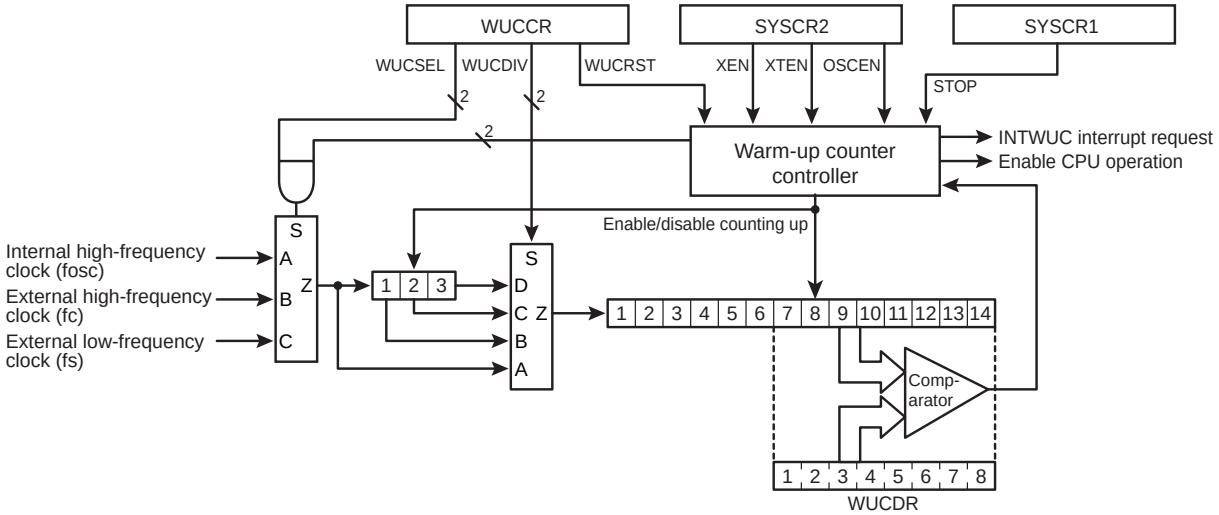


Figure 2-8 Warm-up Counter Circuit

2.3.4.1 Warm-up counter operation when the oscillation is enabled by the hardware

(1) When a power-on reset is released or a reset is released

The warm-up counter serves to secure the time after a power-on reset is released before the supply voltage becomes stable and the time after a reset is released before the oscillation by the high-frequency clock oscillation circuit becomes stable.

When the power is turned on and the supply voltage exceeds the power-on reset release voltage, the warm-up counter reset signal is released. At this time, the CPU and the peripheral circuits are held in the reset state.

A reset signal initializes WUCCR<WUCSEL> to "00" and WUCCR<WUCDIV> to "11", which selects the internal high-frequency clock (fosc) as the input clock to the warm-up counter.

Downloaded from Arrow.com.

When a reset is released for the warm-up counter, the internal high-frequency clock (fosc) is input to the warm-up counter, and the 14-stage counter starts counting the internal high-frequency clock (fosc).

When the upper 8 bits of the warm-up counter become equal to WUCDR, counting is stopped and a reset is released for the CPU and the peripheral circuits.

WUCDR is initialized to 0x66 after reset release, which makes the warm-up time 0x66 × 2^9/fosc [s].

Note: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable.

(2) When the STOP mode is released

The warm-up counter serves to secure the time after the oscillation is enabled by the hardware before the oscillation becomes stable at the release of the STOP mode.

The clock that was used to generate the main system clock when STOP mode was activated is selected as the input clock for the frequency division circuit, regardless of WUCCR<WUCSEL>.

Before the STOP mode is activated, select the division rate of the input clock to the warm-up counter at WUCCR<WUCDIV> and set the warm-up time at WUCDR.

When the STOP mode is released, the 14-stage counter starts counting the input clock selected in the frequency division circuit.

When the upper 8 bits of the warm-up counter become equal to WUCDR, counting is stopped and the operation is restarted by an instruction that follows the STOP mode activation instruction.

Clock that generated the main system clock when the STOP mode was activated	WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
fosc	Don't Care	00	fosc	2^6 / fosc to 255 × 2^6 / fosc
		01	fosc / 2	2^7 / fosc to 255 × 2^7 / fosc
		10	fosc / 2^2	2^8 / fosc to 255 × 2^8 / fosc
		11	fosc / 2^3	2^9 / fosc to 255 × 2^9 / fosc
fc	Don't Care	00	fc	2^6 / fc to 255 × 2^6 / fc
		01	fc / 2	2^7 / fc to 255 × 2^6 / fc
		10	fc / 2^2	2^8 / fc to 255 × 2^8 / fc
		11	fc / 2^3	2^9 / fc to 255 × 2^9 / fc
fs	Don't Care	00	fs	2^6 / fs to 255 × 2^6 / fs
		01	fs / 2	2^7 / fs to 255 × 2^7 / fs
		10	fs / 2^2	2^8 / fs to 255 × 2^8 / fs
		11	fs / 2^3	2^9 / fs to 255 × 2^9 / fs

Note 1: When the operation is switched to the STOP mode during the warm-up for the oscillation enabled by the software, the warm-up counter holds the value at the time, and restarts counting after the STOP mode is released. In this case, the warm-up time at the release of the STOP mode becomes insufficient. Don't switch the operation to the STOP mode during the warm-up for the oscillation enabled by the software.

Note 2: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

Downloaded from Arrow.com.

2.3.4.2 Warm-up counter operation when the oscillation is enabled by the software

The warm-up counter serves to secure the time after the oscillation is enabled by the software before the oscillation becomes stable, at a mode change from NORMAL1 to NORMAL2 or from SLOW1 to SLOW2.

Select the input clock to the frequency division circuit at WUCCR<WUCSEL>.

Select the input clock to the 14-stage counter at WUCCR<WUCDIV>.

After the warm-up time is set at WUCDR, setting SYSCR2<OSCEN>, SYSCR2<XEN> or SYSCR2<XTEN> to "1" allows the stopped oscillation circuit to start oscillation and the 14-stage counter to start counting the selected input clock.

When the upper 8 bits of the counter become equal to WUCDR, an INTWUC interrupt occurs, counting is stopped and the counter is cleared.

Set WUCCR<WUCRST> to "1" to discontinue the warm-up operation.

By setting it to "1", the count-up operation is stopped, the warm-up counter is cleared, and WUCCR<WUCRST> is cleared to "0".

SYSCR2<OSCEN>, SYSCR2<XEN> and SYSCR2<XTEN> hold the values when WUCCR<WUCRST> is set to "1". To restart the warm-up operation, SYSCR2<XEN> or SYSCR2<XTEN> must be cleared to "0".

Note: The warm-up counter starts counting when SYSCR2<OSCEN>, SYSCR2<XEN> or SYSCR2<XTEN> is changed from "0" to "1". The counter will not start counting by writing "1" to SYSCR2<XEN> or SYSCR2<XTEN> when it is in the state of "1".

WUCCR <WUCSEL>	WUCCR <WUCDIV>	Counter input clock	Warm-up time
00	00	fosc	2^6 / fosc to 255 x 2^6 / fosc
	01	fosc / 2	2^7 / fosc to 255 x 2^7 / fosc
	10	fosc / 2^2	2^8 / fosc to 255 x 2^8 / fosc
	11	fosc / 2^3	2^9 / fosc to 255 x 2^9 / fosc
01	00	fc	2^6 / fc to 255 x 2^6 / fc
	01	fc / 2	2^7 / fc to 255 x 2^7 / fc
	10	fc / 2^2	2^8 / fc to 255 x 2^8 / fc
	11	fc / 2^3	2^9 / fc to 255 x 2^9 / fc
10	00	fs	2^6 / fs to 255 x 2^6 / fs
	01	fs / 2	2^7 / fs to 255 x 2^7 / fs
	10	fs / 2^2	2^8 / fs to 255 x 2^8 / fs
	11	fs / 2^3	2^9 / fs to 255 x 2^9 / fs

Note: The clock output from the oscillation circuit is used as the input clock to the warm-up counter. The warm-up time contains errors because the oscillation frequency is unstable until the oscillation circuit becomes stable. Set the sufficient time for the oscillation start property of the oscillator.

2.3.5 Operation mode control circuit

The operation mode control circuit starts and stops the oscillation circuits for the internal high-frequency, external high-frequency and external low-frequency clocks, and switches the main system clock (fm).

There are three operating modes: the single-clock mode, the dual-clock mode and the STOP mode. These modes are controlled by the system control registers (SYSCR1 and SYSCR2).

Figure 2-9 shows the operating mode transition diagram.

Downloaded from Arrow.com.

2.3.5.1 Single-clock mode

Only the gear clock (fcgck) is used for the operation in the single-clock mode.

The main system clock (fm) is generated from the gear clock (fcgck). Therefore, the machine cycle time is 1/fcgck [s].

The gear clock (fcgck) is generated from the high-frequency reference clock (fh).

The high-frequency reference clock (fh) can be selected from the external high-frequency clock (fc) and the internal high-frequency clock (fosc).

When the internal high-frequency clock (fosc) is used as the high-frequency reference clock (fh), pins P00 (XIN) and P01 (XOUT) of the external high-frequency clock oscillation circuit can be used as general-purpose I/O ports.

Before switching the operating mode, be sure to select either the external high-frequency clock (fc) or the internal high-frequency clock (fosc) and then stop either of the high-frequency clocks not to be used. If a mode transition is made with both the external and internal high-frequency clocks enabled, the transition may not be performed properly.

For how to switch the high-frequency reference clock (fh), refer to "(1) High-frequency reference clock (fh)".

In the single-clock mode, pins P02 (XTIN) and P03 (XTOUT) of the external low-frequency clock oscillation circuit can be used as general-purpose I/O ports.

(1) NORMAL1 mode

In this mode, the CPU core and the peripheral circuits operate using the gear clock (fcgck).

After reset release, the NORMAL1 mode becomes active and the internal high-frequency clock (fosc) is used as the high-frequency reference clock (fh).

(2) IDLE1 mode

In this mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck).

The IDLE1 mode is activated by setting SYSCR2<IDLE> to "1" in the NORMAL1 mode.

When the IDLE1 mode is activated, the CPU and the watchdog timer stop.

When the interrupt latch enabled by the interrupt enable register EFR becomes "1", the IDLE1 mode is released to the NORMAL1 mode.

When the IMF (interrupt master enable flag) is "1" (interrupts enabled), the operation returns normal after the interrupt processing is completed.

When the IMF is "0" (interrupts disabled), the operation is restarted by the instruction that follows the IDLE1 mode activation instruction.

(3) IDLE0 mode

In this mode, the CPU and the peripheral circuits stop, except the oscillation circuits and the time base timer.

In the IDLE0 mode, the peripheral circuits stop in the states when the IDLE0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the IDLE0 mode, refer to the section of each peripheral circuit.

The IDLE0 mode is activated by setting SYSCR2<TGHALT> to "1" in the NORMAL1 mode.

Downloaded from Arrow.com.

When the IDLE0 mode is activated, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer.

When the falling edge of the source clock selected at TBTCR<TBTCK> is detected, the IDLE0 mode is released, the timing generator starts the clock supply to all the peripheral circuits and the NORMAL1 mode is restored.

Note that the IDLE0 mode is activated and restarted, regardless of the setting of TBTCR<TBTEN>.

When the IDLE0 mode is activated with TBTCR<TBTEN> set at "1", the INTTBT interrupt latch is set after the NORMAL mode is restored.

When the IMF is "1" and the EF5 (the individual interrupt enable flag for the time base timer) is "1", the operation returns normal after the interrupt processing is completed.

When the IMF is "0" or when the IMF is "1" and the EF5 (the individual interrupt enable flag for the time base timer) is "0", the operation is restarted by the instruction that follows the IDLE0 mode activation instruction.

2.3.5.2 Dual-clock mode

The gear clock (fcgck) and the external low-frequency clock (fs) are used for the operation in the dual-clock mode.

The main system clock (fm) is generated from the gear clock (fcgck) in the NORMAL2 or IDLE2 mode, and generated from the clock that is a quarter of the low-frequency clock (fs) in the SLOW1/2 or SLEEP0/1 mode. Therefore, the machine cycle time is 1/fcgck [s] in the NORMAL2 or IDLE2 mode and is 4/fs [s] in the SLOW1/2 or SLEEP0/1 mode.

Pins P02 (XTIN) and P03 (XTOUT) are used for the low-frequency clock oscillation circuit. (These pins cannot be used as I/O ports in the dual-clock mode.)

The gear clock (fcgck) is generated from the high-frequency reference clock (fh).

The high-frequency reference clock (fh) can be selected from the external high-frequency clock (fc) and the internal high-frequency clock (fosc).

When the internal high-frequency clock (fosc) is used as the high-frequency reference clock (fh), pins P00 (XIN) and P01(XOUT) of the external high-frequency clock oscillation circuit can be used as general-purpose I/O ports.

Before switching the operating mode, be sure to select either the external high-frequency cock (fc) or the internal high-frequency clock (fosc) and then stop either of the high-frequency clocks not to be used. If a mode transition is made with both the external and internal high-frequency clocks enabled, the transition may not be performed properly.

For how to switch the high-frequency reference clock (fh), refer to "(1) High-frequency reference clock (fh)".

SYSCR1<OSCSEL> cannot be changed when SYSCR1<SYSCK> is "1". Therefore, when switching between the SLOW1 and SLOW2 modes, the high-frequency reference clock (fh) must be set in advance in the NORMAL1 or NORMAL2 mode.

TLCS-870/C1 Series devices start up in the single-clock mode after reset release. To use the dual-clock mode, activate the low-frequency clock by software.

(1) NORMAL2 mode

In this mode, the CPU core operates using the gear clock (fcgck), and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

Downloaded from Arrow.com.

(2) SLOW2 mode

In this mode, the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2<SYSCK> to switch the operation mode from NORMAL2 to SLOW2 or from SLOW2 to NORMAL2.

In the SLOW2 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(3) SLOW1 mode

In this mode, the high-frequency clock oscillation circuit stops operation and the CPU core and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

This mode requires less power to operate the high-frequency clock oscillation circuit than in the SLOW2 mode.

In the SLOW mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLOW mode, refer to the section of each peripheral circuit.

Set SYSCR2<XEN> to switch the operation between the SLOW1 and SLOW2 modes.

In the SLOW1 or SLEEP1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

(4) IDLE2 mode

In this mode, the CPU and the watchdog timer stop and the peripheral circuits operate using the gear clock (fcgck) or the clock that is a quarter of the low-frequency clock (fs).

The IDLE2 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the NORMAL2 mode after this mode is released.

(5) SLEEP1 mode

In this mode, the high-frequency clock oscillation circuit stops operation, the CPU and the watchdog timer stop, and the peripheral circuits operate using the clock that is a quarter of the low-frequency clock (fs).

In the SLEEP1 mode, some peripheral circuits become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP1 mode, refer to the section of each peripheral circuit.

The SLEEP1 mode can be activated and released in the same way as for the IDLE1 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLOW1 or SLEEP1 mode, outputs of the prescaler and stages 1 to 8 of the divider stop.

Note: A transition to SLEEP0 mode or SLEEP1 mode must be performed on the RAM or the Shadow RAM. Note that in these processes interrupts must be prohibited; interrupt vectors and routines must be placed on RAM or Shadow RAM.

Downloaded from Arrow.com.

(6) SLEEP0 mode

In this mode, the high-frequency clock oscillation circuit stops operation, the time base timer operates using the clock that is a quarter of the low-frequency clock (fs), and the core and the peripheral circuits stop.

In the SLEEP0 mode, the peripheral circuits stop in the states when the SLEEP0 mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the SLEEP0 mode, refer to the section of each peripheral circuit.

The SLEEP0 mode can be activated and released in the same way as for the IDLE0 mode. The operation returns to the SLOW1 mode after this mode is released.

In the SLEEP0 mode, the CPU stops and the timing generator stops the clock supply to the peripheral circuits except the time base timer.

Note: A transition to SLEEP0 mode or SLEEP1 mode must be performed on the RAM or the Shadow RAM. Note that in these processes interrupts must be prohibited; interrupt vectors and routines must be placed on RAM or Shadow RAM.

2.3.5.3 STOP mode

In this mode, all the operations in the system, including the oscillation circuits, are stopped and the internal states in effect before the system was stopped are held with low power consumption.

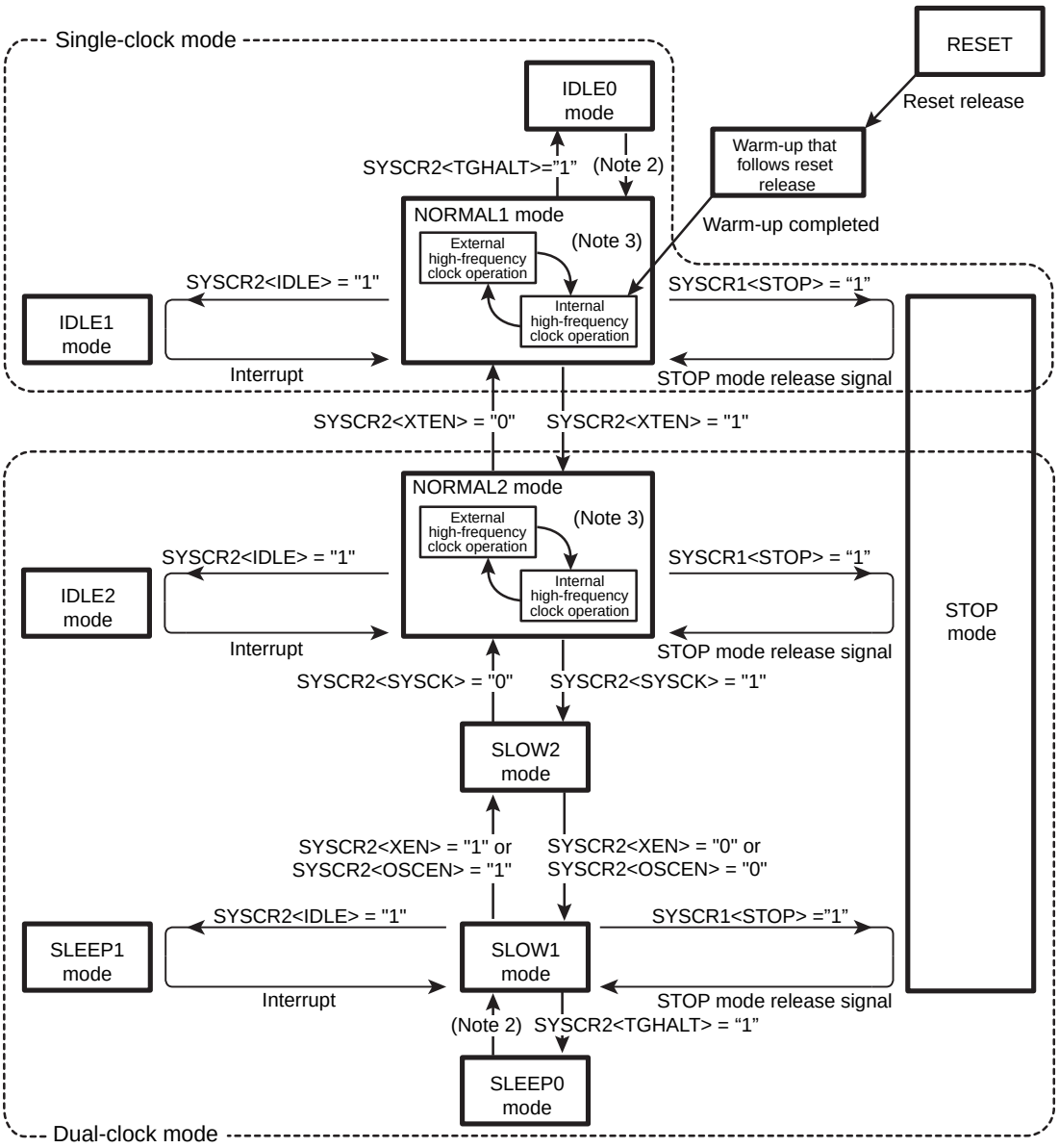
In the STOP mode, the peripheral circuits stop in the states when the STOP mode is activated or become the same as the states when a reset is released. For operations of the peripheral circuits in the STOP mode, refer to the section of each peripheral circuit.

The STOP mode is activated by setting SYSCR1<STOP> to "1".

The STOP mode is released by the STOP mode release signals. After the warm-up time has elapsed, the operation returns to the mode that was active before the STOP mode, and the operation is restarted by the instruction that follows the STOP mode activation instruction.

Downloaded from Arrow.com.

2.3.5.4 Transition of operation modes



Note 1: The NORMAL1 and NORMAL2 modes are generically called the NORMAL mode; the SLOW1 and SLOW2 modes are called the SLOW mode; the IDLE0, IDLE1 and IDLE2 modes are called the IDLE mode; and the SLEEP0 and SLEEP1 are called the SLEEP mode.

Note 2: The mode is released by the falling edge of the source clock selected at TBTCR<TBTCK>.

Note 3: Switching between the internal high-frequency clock and the external high-frequency clock must be done during the NORMAL1 or NORMAL2 mode. For details, refer to "(1) High-frequency reference clock (fh)".

Figure 2-9 Operation Mode Transition Diagram

Downloaded from Arrow.com.

Table 2-5 Operation Modes and Conditions

Operation mode		Oscillation circuit: High-frequency reference clock (fh)	Oscillation circuit: Low-frequency clock (fs)	CPU core	Watchdog timer	Time base timer	AD converter	Other peripheral circuits	Machine cycle time
Single clock	RESET	Oscillation	Stop	Reset	Reset	Reset	Reset	Reset	1 / fcgck [s]
	NORMAL1			Operate	Operate	Operate	Operate	Operate	
	IDLE1			Stop	Stop				
	IDLE0						Stop	Stop	
	STOP	Stop				Stop			–
Dual clock	NORMAL2	Oscillation	Oscillation	Operate with the high frequency	Operate with the high / low frequency	Operate	Operate	Operate	1 / fcgck [s]
	IDLE2			Stop	Stop				
	SLOW2			Operate with the low frequency	Operate with the low frequency		Stop		4/ fs [s]
	SLOW1	Stop		Operate with the low frequency	Operate with the low frequency				
	SLEEP1			Stop	Stop				
	SLEEP0							Stop	
	STOP		Stop			Stop			–

2.3.6 Operation Mode Control

2.3.6.1 STOP mode

The STOP mode is controlled by system control register 1 (SYSCR1) and the STOP mode release signals.

(1) Start the STOP mode

The STOP mode is started by setting SYSCR1<STOP> to "1". In the STOP mode, the following states are maintained:

1. Both the internal (or external) high-frequency and external low-frequency clock oscillation circuits stop oscillation and all internal operations are stopped.

2. The data memory, the registers and the program status word are all held in the states in effect before STOP mode was started. The port output latch is determined by the value of SYSCR1<OUTEN>.

3. The prescaler and the divider of the timing generator are cleared to "0".

4. The program counter holds the address of the instruction 2 ahead of the instruction (e.g., [SET (SYSCR1).7]) which started the STOP mode.

Downloaded from Arrow.com.

(2) Release the STOP mode

The STOP mode is released by the following STOP mode release signals. It is also released by a reset by the $\overline{\text{RESET}}$ pin, a power-on reset and a reset by the voltage detection circuits. When a reset is released, the warm-up starts. After the warm-up operation and copying of the shadow RAM are completed, the NORMAL1 mode becomes active.

1. Release by the STOP pin
2. Release by key-on wakeup
3. Release by the voltage detection circuits

Note: During the STOP period (from the start of the STOP mode to the end of the warm-up), due to changes in the external interrupt pin signal, interrupt latches may be set to "1" and interrupts may be accepted immediately after the STOP mode is released. Before starting the STOP mode, therefore, disable interrupts. Also, before enabling interrupts after STOP mode is released, clear unnecessary interrupt latches.

1. Release by the $\overline{\text{STOP}}$ pin

Release the $\overline{\text{STOP}}$ mode by using the STOP pin.

The STOP mode release by the STOP pin includes the level-sensitive release mode and the edge-sensitive release mode, either of which can be selected at SYSCR1<RELM>.

The $\overline{\text{STOP}}$ pin is also used as the P47 port and the $\overline{\text{INT5}}$ (external interrupt input 5) pin.

- Level-sensitive release mode

The STOP mode is released by setting the $\overline{\text{STOP}}$ pin high.

Setting SYSCR1<RELM> to "1" selects the level-sensitive release mode.

This mode is used for the capacitor backup when the main power supply is cut off and the long term battery backup.

Even if an instruction for starting the STOP mode is executed while the $\overline{\text{STOP}}$ pin input is high, the STOP mode does not start. Thus, to start the STOP mode in the level-sensitive release mode, it is necessary for the program to first confirm that the $\overline{\text{STOP}}$ pin input is low.

This can be confirmed by testing the port by the software or using interrupts

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

Example: Starting the STOP mode from NORMAL mode after testing port.
(Warm-up time at release of the STOP mode is about 300μs at fc= 10MHz.)

```
        LD      (SYSCR1), 0x40          ;Sets up the level-sensitive release mode
SSTOPH: TEST    (P4PRD). 7              ;Wait until STOP pin becomes L level.
        J       F, code_addr(SSTOPH)
        LD      (WUCCR), 0x01           ;WUCCR<WUCDIV> = 00 (No division) (Note)
        LD      (WUCDR),0x2F            ;Sets the warm-up time
                                        ;300μs / 6.4μs = 46.9 → round up to 0x2F
        DI                              ;IMF = 0
        SET     (SYSCR1).7              ;Starts the STOP mode
```

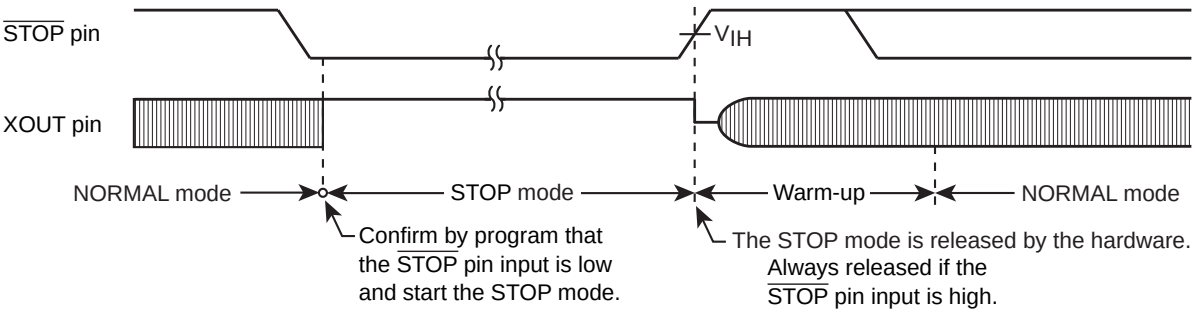
Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

Downloaded from Arrow.com.

Example: Starting the STOP mode from the SLOW mode with an INT5 interrupt
(Warm-up time at release of the STOP mode is about 450ms at fs=32.768 kHz.)

```
PINT5:   TEST   (P4PRD). 7          ;To reject noise, the STOP mode does not start
         J      F, code_addr(SINT5) ;if the STOP pin input is high.
         LD     (SYSCR1), 0x40      ;Sets up the level-sensitive release mode
         LD     (WUCCR), 0x03       ;WUCCR<WUCDIV> = 00 (No division) (Note)
         LD     (WUCDR),0xE8        ;Sets the warm-up time
                                    ;450 ms/1.953 ms = 230.4 → round up to 0xE8
         DI                         ;IMF = 0
         SET    (SYSCR1).7          ;Starts the STOP mode
SINT5:   RETI
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.



Note: Even if the STOP pin input returns to low after the warm-up starts, the STOP mode is not restarted.

Figure 2-10 Level-sensitive Release Mode (Example when the high-frequency clock oscillation circuit is selected)

- Edge-sensitive release mode

In this mode, the STOP mode is released at the rising edge of the STOP pin input.

Setting SYSCR1<RELM> to "0" selects the edge-sensitive release mode.

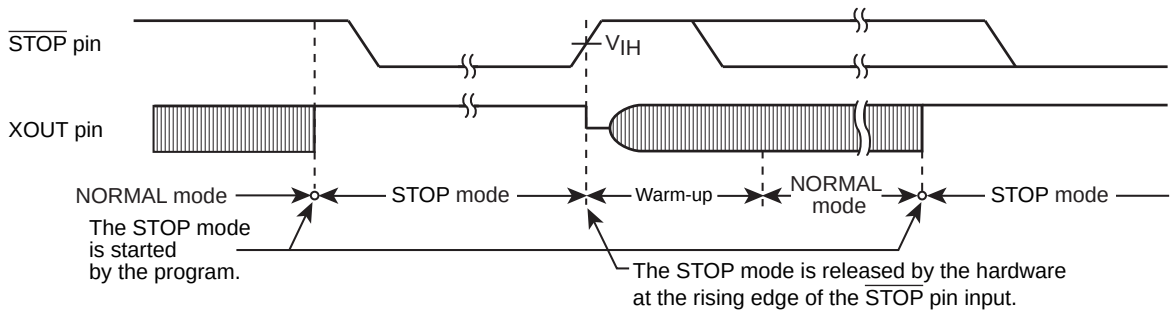
This is used in applications where a relatively short program is executed repeatedly at periodic intervals. This periodic signal (for example, a clock from a low-power consumption oscillator) is input to the STOP pin. In the edge-sensitive release mode, the STOP mode is started even when the STOP pin input is high

Example: Starting the STOP mode from the NORMAL mode
(Warm-up time at release of the STOP mode is about 200μs at fc=10 MHz.)

```
         LD     (WUCCR),0x01        ;WUCCR<WUCDIV> = 00 (No division) (Note)
         LD     (WUCDR),0x20        ;Sets the warm-up time
                                    ;200μs / 6.4μs = 31.25 → round up to 0x20
         DI                         ;IMF = 0
         LD     (SYSCR1) , 0x80     ;Starts the STOP mode with the edge-sensitive release mode selected
```

Note: When the STOP mode is released, the warm-up counter source clock automatically changes to the clock that generated the main system clock when the STOP mode was started, regardless of WUCCR<WUCSEL>.

Downloaded from Arrow.com.



Note: If the rising edge is input to the $\overline{\text{STOP}}$ pin within 1 machine cycle after SYSCR1<STOP> is set to "1", the STOP mode will not be released.

Figure 2-11 Edge-sensitive Release Mode (Example when the high-frequency clock oscillation circuit is selected)

2. Release by the key-on wakeup

   The STOP mode is released by inputting the prescribed level to the key-on wakeup pin.

   The level to release the STOP mode can be selected from "H" and "L".

   For release by the key-on wakeup, refer to section "Key-on Wakeup".

   Note: If the key-on wakeup pin input becomes the opposite level to the release level after the warm-up starts, the STOP mode is not restarted.

3. Release by the voltage detection circuits

   The STOP mode is released by the supply voltage detection by the voltage detection circuits.

   If the voltage detection operation mode of the voltage detection circuits is set to "Generates a voltage detection reset signal", the STOP mode is released and a reset is applied as soon as the supply voltage becomes lower than the detection voltage.

   When the supply voltage becomes equal to or higher than the detection voltage of the voltage detection circuits, the reset is released and the warm-up starts. After the warm-up is completed, the NORMAL1 mode becomes active.

   For details, refer to the chapter on the voltage detection circuit.

   Note: If the supply voltage becomes equal to or higher than the detection voltage within 1 machine cycle after SYSCR1<STOP> is set to "1", the STOP mode will not be released.

(3) STOP mode release operation

The STOP mode is released in the following sequence:

1. Oscillation starts. For the oscillation start operation in each mode, refer to "Table 2-6 Oscillation Start Operation at Release of the STOP Mode".

2. Warm-up is executed to secure the time required to stabilize oscillation. The internal operations remain stopped during warm-up. The warm-up time is set by the warm-up counter, depending on the oscillator characteristics.

3. After the warm-up time has elapsed, the normal operation is restarted by the instruction that follows the STOP mode start instruction. At this time, the prescaler and the divider of the timing generator are cleared to "0".

Downloaded from Arrow.com.

Note: When the STOP mode is released with a low hold voltage, the following cautions must be observed.

The supply voltage must be at the operating voltage level before releasing the STOP mode. The $\overline{\text{RESET}}$ pin input must also be "H" level, rising together with the supply voltage. In this case, if an external time constant circuit has been connected, the $\overline{\text{RESET}}$ pin input voltage will increase at a slower pace than the power supply voltage. At this time, there is a danger that a reset may occur if the input voltage level of the $\overline{\text{RESET}}$ pin drops below the non-inverting high-level input voltage (Hysteresis input).

Table 2-6 Oscillation Start Operation at Release of the STOP Mode

Operation mode before the STOP mode is started		High-frequency reference clock	Low-frequency reference clock	Oscillation start operation after release
Single-clock mode	NORMAL1	Internal high-frequency clock	-	The internal high-frequency clock oscillation circuit starts oscillation. The external high-frequency clock oscillation circuit is inactive. The external low-frequency clock oscillation circuit is inactive.
		External high-frequency clock	-	The internal high-frequency clock oscillation circuit is inactive. The external high-frequency clock oscillation circuit starts oscillation. The external low-frequency clock oscillation circuit is inactive.
Dual-clock mode	NORMAL2	Internal high-frequency clock	External low-frequency clock	The internal high-frequency clock oscillation circuit starts oscillation. The external high-frequency clock oscillation circuit is inactive. The external low-frequency clock oscillation circuit starts oscillation.
		External high-frequency clock	External low-frequency clock	The internal high-frequency clock oscillation circuit is inactive. The external high-frequency clock oscillation circuit starts oscillation. The external low-frequency clock oscillation circuit starts oscillation.
	SLOW1	-	External low-frequency clock	The internal high-frequency clock oscillation circuit is inactive. The external high-frequency clock oscillation circuit is inactive. The external low-frequency clock oscillation circuit starts oscillation.

2.3.6.2 IDLE1/2 and SLEEP1 modes

The IDLE1/2 and SLEEP1 modes are controlled by the system control register 2 (SYSCR2) and maskable interrupts. The following states are maintained during these modes.

1. The CPU and the watchdog timer stop their operations. The peripheral circuits continue to operate.

2. The data memory, the registers, the program status word and the port output latches are all held in the status in effect before IDLE1/2 or SLEEP1 mode was started.

3. The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE1/2 or SLEEP1 mode.

Note: A transition to SLEEP0 mode or SLEEP1 mode must be performed on the RAM or the Shadow RAM. Note that in these processes interrupts must be prohibited; interrupt vectors and routines must be placed on RAM or Shadow RAM.

Downloaded from Arrow.com.

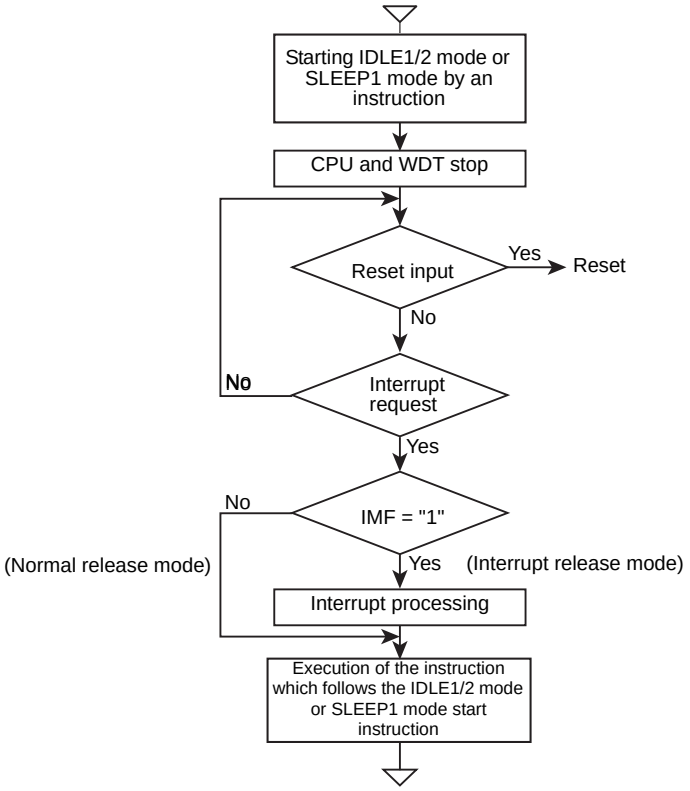


Figure 2-12 IDLE1/2 and SLEEP 1 Modes

Downloaded from Arrow.com.

(1) Start the IDLE1/2 and SLEEP1 modes

After the interrupt master enable flag (IMF) is set to "0", set the individual interrupt enable flag (EF) to "1", which releases IDLE1/2 and SLEEP1 modes.

To start the IDLE1/2 or SLEEP1 mode, set SYSCR2<IDLE> to "1".

If the release condition is satisfied when it is attempted to start the IDLE1/2 or SLEEP1 mode, SYSCR2<IDLE> remains cleared and the IDLE1/2 or SLEEP1 mode will not be started.

Note 1: When a watchdog timer interrupt is generated immediately before the IDLE1/2 or SLEEP1 mode is started, the watchdog timer interrupt will be processed but the IDLE1/2 or SLEEP1 mode will not be started.

Note 2: Before starting the IDLE1/2 or SLEEP1 mode, enable the interrupt request signals to be generated to release the IDLE1/2 or SLEEP1 mode and set the individual interrupt enable flag.

(2) Release the IDLE1/2 and SLEEP1 modes

The IDLE1/2 and SLEEP1 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (IMF). After releasing IDLE1/2 or SLEEP1 mode, SYSCR2<IDLE> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE1/2 or SLEEP1 mode.

The IDLE1/2 and SLEEP1 modes are also released by a reset by the $\overline{\text{RESET}}$ pin. After releasing the reset, the warm-up starts. After the warm-up operation and copying of the shadow RAM are completed, the NORMAL1 mode becomes active.

- Normal release mode (IMF = "0")

  The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (EF) is "1". The operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction. Normally, the interrupt latch (IL) of the interrupt source used for releasing must be cleared to "0" by load instructions.

- Interrupt release mode (IMF = "1")

  The IDLE1/2 or SLEEP1 mode is released when the interrupt latch enabled by the individual interrupt enable flag (EF) is "1". After the interrupt is processed, the operation is restarted by the instruction that follows the IDLE1/2 or SLEEP1 mode start instruction.

2.3.6.3 IDLE0 and SLEEP0 modes

The IDLE0 and SLEEP0 modes are controlled by the system control register 2 (SYSCR2) and the time base timer control register (TBTCR). The following states are maintained during the IDLE0 and SLEEP0 modes:

- The timing generator stops the clock supply to the peripheral circuits except the time base timer.
- The data memory, the registers, the program status word and the port output latches are all held in the states in effect before the IDLE0 or SLEEP0 mode was started.
- The program counter holds the address of the instruction 2 ahead of the instruction which starts the IDLE0 or SLEEP0 mode.

Note: A transition to SLEEP0 mode or SLEEP1 mode must be performed on the RAM or the Shadow RAM. Note that in these processes interrupts must be prohibited; interrupt vectors and routines must be placed on RAM or Shadow RAM.

Downloaded from Arrow.com.

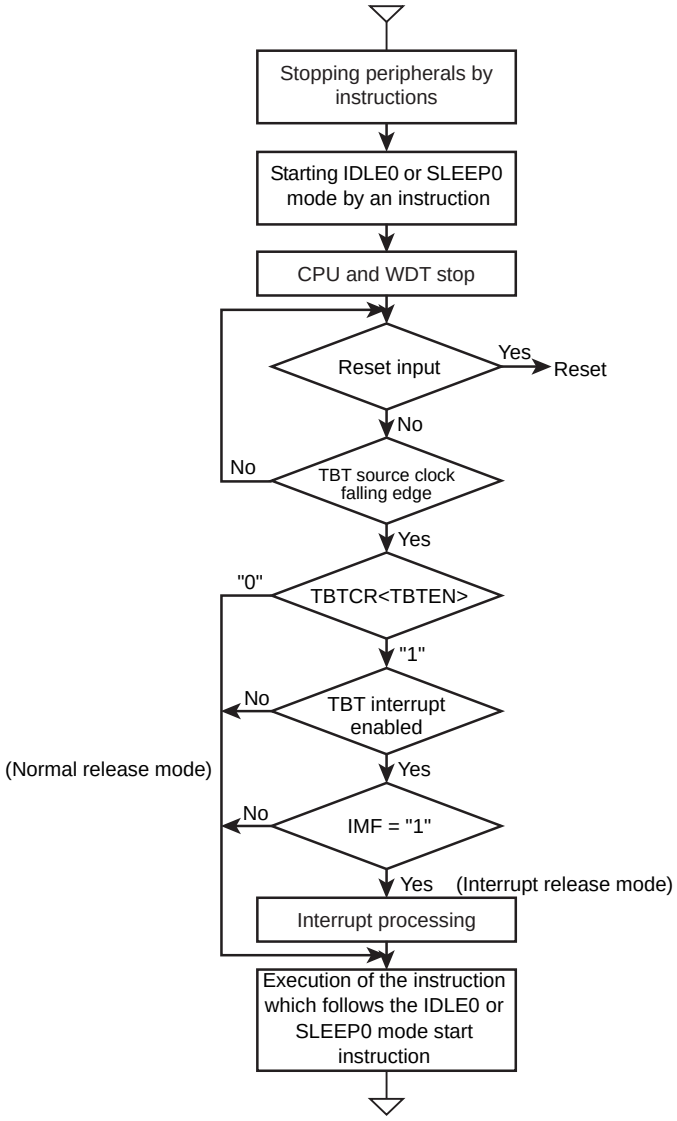


Figure 2-13 IDLE0 and SLEEP0 Modes

- Start the IDLE0 and SLEEP0 modes

  Stop (disable) the peripherals such as a timer counter.

  To start the IDLE0 or SLEEP0 mode, set SYSCR2<TGHALT> to "1".

- Release the IDLE0 and SLEEP0 modes

  The IDLE0 and SLEEP0 modes include a normal release mode and an interrupt release mode. These modes are selected at the interrupt master enable flag (IMF), the individual interrupt enable flag (EF5) for the time base timer and TBTCR<TBTEN>. After releasing the IDLE0 or SLEEP0 mode, SYSCR2<TGHALT> is automatically cleared to "0" and the operation mode is returned to the mode preceding the IDLE0 or SLEEP0 mode. If TBTCR<TBTEN> has been set at "1", the INTTBT interrupt latch is set.

  The IDLE0 and SLEEP0 modes are also released by a reset by the $\overline{\text{RESET}}$ pin. When a reset is released, the warm-up starts. After the warm-up operation and copying of the shadow RAM are completed, the NORMAL1 mode becomes active.

Downloaded from Arrow.com.

(1) Normal release mode (IMF, EF5, TBTCR<TBTEN> = "0")

The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR<TBTCK> is detected. After the IDLE0 or SLEEP0 mode is released, the operation is restarted by the instruction that follows the IDLE0 or SLEEP0 mode start instruction.

When TBTCR<TBTEN> is "1", the time base timer interrupt latch is set.

(2) Interrupt release mode (IMF, EF5, TBTCR<TBTEN> = "1")

The IDLE0 or SLEEP0 mode is released when the falling edge of the source clock selected at TBTCR<TBTCK> is detected. After the release, the INTTBT interrupt processing is started.

Note 1: The IDLE0 or SLEEP0 mode is released to the NORMAL1 or SLOW1 mode by the asynchronous internal clock selected at TBTCR<TBTCK>. Therefore, the period from the start to the release of the mode may be shorter than the time specified at TBTCR<TBTCK>.

Note 2: When a watchdog timer interrupt is generated immediately before the IDLE0 or SLEEP0 mode is started, the watchdog timer interrupt will be processed but the IDLE0 or SLEEP0 mode will not be started.

2.3.6.4 SLOW mode

The SLOW mode is controlled by system control register 2 (SYSCR2).

(1) Switching from the NORMAL2 mode to the SLOW1 mode

Set SYSCR2<SYSCK> to "1".

When a maximum of 2/fcgck + 10/fs [s] has elapsed since SYSCR2<SYSCK> is set to "1", the main system clock (fm) is switched to fs/4.

After switching, wait for 2 machine cycles or longer, and then clear SYSCR2<XEN> to "0" to turn off the high-frequency clock oscillator.

If the oscillation of the low-frequency clock (fs) is unstable, confirm the stable oscillation at the warm-up counter before implementing the procedure described above.

Note 1: Be sure to follow this procedure to switch the operation from the NORMAL2 mode to the SLOW1 mode.

Note 2: It is also possible to allow the basic clock for the high-frequency clock to oscillate continuously to return to NORMAL2 mode. However, be sure to turn off the oscillation of the basic clock for the high-frequency clock when the STOP mode is started from the SLOW mode.

Note 3: After switching SYSCR2<SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2<XEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 4: When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the basic clock (fs) for the low-frequency clock. For the synchronization, fm is stopped for a period of 10/fs or shorter.

Downloaded from Arrow.com.

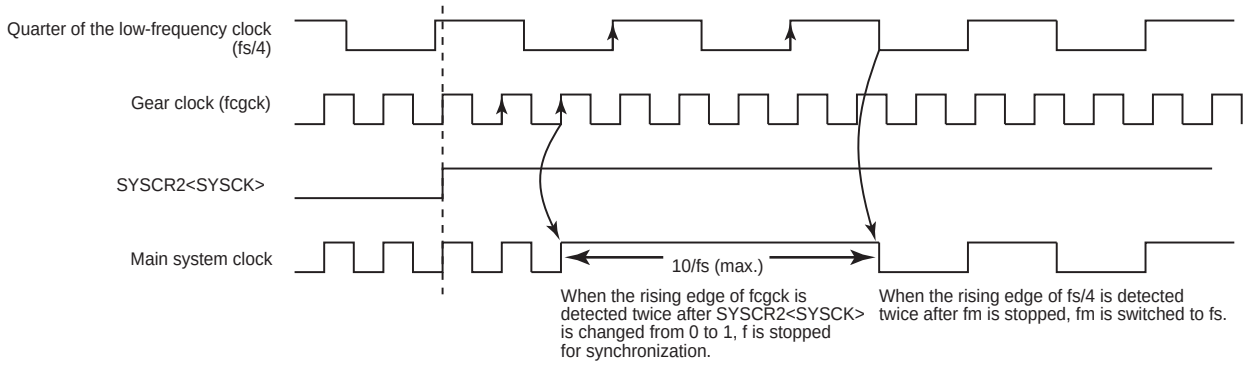


Figure 2-14 Switching of the Main System Clock (fm) (Switching from fcgck to fs/4)

Example 1: Switching from the NORMAL2 mode to the SLOW1 mode (when fc is used as the basic clock for the high-frequency clock)

```
SET     (SYSCR2).4          ;SYSCR2<SYSCK> = 1
                            ;(Switches the main system clock to the basic clock for the
                            ;low-frequency clock for the SLOW2 mode)
NOP                         ;Waits for 2 machine cycles
NOP
CLR     (SYSCR2).6          ;SYSCR2<XEN> = 0
                            ;(Turns off the high-frequency clock oscillation circuit)
```

Example 2: While operating with the external high-frequency clock, switching to the SLOW1 mode after the stable oscillation of the external low-frequency clock oscillation circuit is confirmed at the warm-up counter (fs=32.768kHz, warm-up time = about 100 ms)

```
; #### Initialize routine ####
SET     (P0FC).2            ;P0FC2 = 1 (Uses P02/03 as oscillators)
¦
¦
LD      (WUCCR), 0x02       ;WUCCR<WUCDIV> = 00 (No division)
                            ;WUCCR<WUCSEL> = 1 (Selects fs as the source clock)
LD      (WUCDR), 0x33       ;Sets the warm-up time
                            ;(Determines the time depending on the oscillator characteristics)
                            ;100 ms/1.95 ms = 51.2 → round up to 0x33
SET     (EIRL).4            ;Enables INTWUC interrupts
SET     (SYSCR2).5          ;SYSCR2<XTEN> = 1
                            ;(Starts the low-frequency clock oscillation and starts the warm-up
                            ;counter)
¦
; #### Interrupt service routine of warm-up counter interrupts ####
PINTWUC:  SET   (SYSCR2).4  ;SYSCR2<SYSCK> = 1
                            ;(Switches the main system clock to the low-frequency clock)
NOP                         ;Waits for 2 machine cycles
NOP
CLR     (SYSCR2).6          ;SYSCR2<XEN> = 0
                            ;(Turns off the high-frequency clock oscillation circuit)
RETI
¦
```


Downloaded from Arrow.com.

```
VINTWUC:    DW    code_addr(PINTWUC)        ;INTWUC vector table
```

(2) Switching from the SLOW1 mode to the NORMAL1 mode

Set SYSCR2<OSCEN> or SYSCR2<XEN> to "1" to enable the high-frequency reference clock (fh). Confirm at the warm-up counter that the oscillation of the reference clock for the high-frequency clock has stabilized, and then clear SYSCR2<SYSCK> to "0".

When a maximum of 8/fs + 2.5/fcgck [s] has elapsed since SYSCR2<SYSCK> is cleared to "0", the main system clock (fm) is switched to fcgck.

After switching, wait for 2 machine cycles or longer, and then clear SYSCR2<XTEN> to "0" to turn off the external low-frequency clock oscillator.

The SLOW mode is also released by a reset by the $\overline{\text{RESET}}$ pin. When a reset is released, the warm-up starts. After the warm-up operation and copying of the shadow RAM are completed, the NORMAL1 mode becomes active.

Note 1: Be sure to follow this procedure to switch the operation from the SLOW1 mode to the NORMAL1 mode.

Note 2: After switching SYSCR2<SYSCK>, be sure to wait for 2 machine cycles or longer before clearing SYSCR2<XTEN> to "0". Clearing it within 2 machine cycles causes a system clock reset.

Note 3: When the main system clock (fm) is switched, the gear clock (fcgck) is synchronized with the clock that is a quarter of the basic clock (fs) for the low-frequency clock. For the synchronization, fm is stopped for a period of 2.5/fcgck [s] or shorter.

Note 4: SYSCR1<OSCSEL> should be set while SYSCR2<SYSCK> is "0" (during the NORMAL1 or NORMAL 2 mode). Writing to SYSCR1<OSCSEL> while SYSCR2<SYSCK> is "1" (during the SLOW1 or SLOW2 mode) has no effect.

Note 5: When P0FC0 is "0", setting SYSCR2<XEN> to "1" causes a system clock reset.

Note 6: When SYSCR2<XEN> is set at "1", writing "1" to SYSCR2<XEN> does not cause the warm-up counter to start counting the source clock.

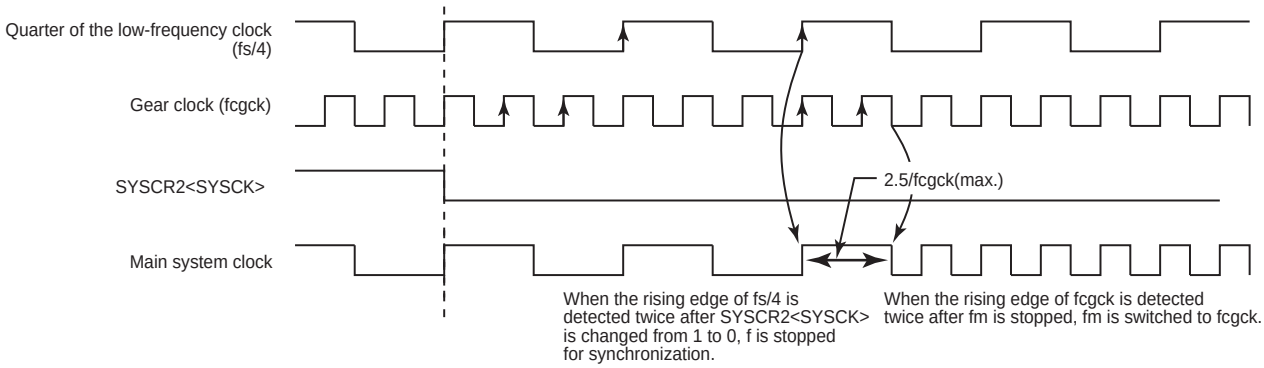


Figure 2-15 Switching the Main System Clock (fm) (Switching from fs/4 to fcgck)

Example : Switching from the SLOW1 mode to the NORMAL1 mode after the stability of the external high-frequency clock oscillation circuit is confirmed at the warm-up counter (fc = 10 MHz, warm-up time = 4.0 ms)

```
; #### Initialize routine ####
SET     (P0FC).2                ;P0FC2 = 1 (Uses P02/03 as oscillators)
¦
¦
LD      (WUCCR), 0x09           ;WUCCR<WUCDIV> = 10 (Divided by 2)
                                ;WUCCR<WUCSEL> = 0 (Selects fc as the source clock)
LD      (WUCDR), 0x9D           ;Sets the warm-up time
                                ;(Determine the time depending on the frequency and the oscillator
```


Downloaded from Arrow.com.

```
                                                    ;characteristics)
                                                    ;4ms / 25.6us = 156.25 → round up to 0x9D
        SET     (EIRL). 4                           ;Enables INTWUC interrupts
        SET     (SYSCR2) .6                         ;SYSCR2<XEN> = 1
                                                    ;(Starts the oscillation of the high-frequency clock oscillation circuit)
        ¦
        ; #### Interrupt service routine of warm-up counter interrupts ####
PINTWUC:CLR     (SYSCR2). 4                         ;SYSCR2<SYSCK> = 0
                                                    ;(Switches the main system clock to the gear clock)
        NOP                                         ;Waits for 2 machine cycles
        NOP
        CLR     (SYSCR2). 5                         ;SYSCR2<XTEN> = 0
                                                    ;(Turns off the external low-frequency clock oscillation circuit)
        RETI
        ¦
VINTWUC:DW      code_addr(PINTWUC)                  ;INTWUC vector table
```


Downloaded from Arrow.com.

2.4 Reset Control Circuit

The reset circuit controls the external and internal factor resets and initializes the system.

2.4.1 Configuration

The reset control circuit consists of the following reset signal generation circuits:

1. External reset input (external factor)
2. Power-on reset (internal factor)
3. Voltage detection reset (internal factor)
4. Watchdog timer reset (internal factor)
5. System clock reset (internal factor)

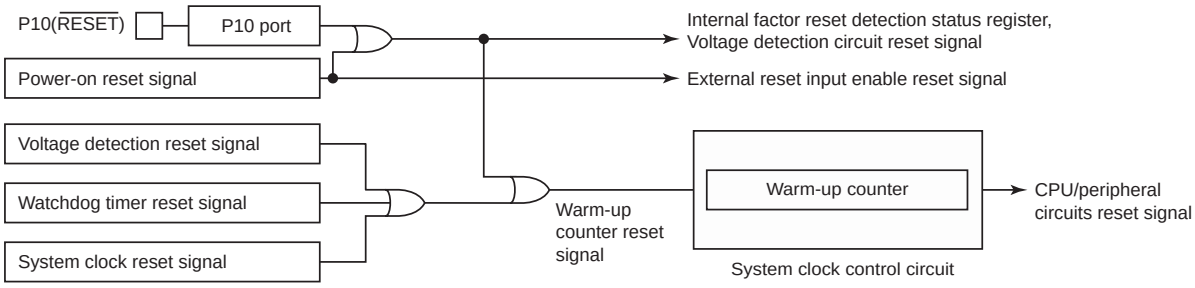


Figure 2-16 Reset Control Circuit

2.4.2 Control

The reset control circuit is controlled by system control register 3 (SYSCR3), system control register 4 (SYSCR4), system control status register (SYSSR4) and the internal factor reset detection status register (IRSTSR).

System control register 3

SYSCR3 (0x00FDE)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	(RVCTR)	(RAREA)	RSTDIS
Read/Write	R	R	R	R	R	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1: The enabled SYSCR3<RSTDIS> is initialized by a power-on reset only, and cannot be initialized by an external reset input or internal factor reset. The value written in SYSCR3 is reset by a power-on reset, external reset input or internal factor reset.

Note 2: The value of SYSCR3<RSTDIS> is invalid until 0xB2 is written into SYSCR4.

Note 3: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL1 mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 4: Bits 7 to 3 of SYSCR3 are read as "0".

Note 5: When using the voltage detection circuit, must protect the RESET pin from external noise. If the RESET pin recognizes as an external reset input, the voltage detection circuit will be initialized and the voltage detection reset signal is released. CPU and the peripheral circuit start operation. At this time, if supply voltage is below in the level of recommended voltage, this may cause malfunction.

Downloaded from Arrow.com.

System control register 4

SYSCR4 (0x00FDF)

	7	6	5	4	3	2	1	0
Bit Symbol	SYSCR4							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

SYSCR4	Writes the SYSCR3 data control code.	0xB2 :	Enables the contents of SYSCR3<RSTDIS>
		0xD4 :	Enables the contents of SYSCR3<RAREA> and SYSCR3 <RVCTR>
		0x71 :	Enables the contents of IRSTSR<FCLR>
		Others :	Invalid

Note 1: SYSCR4 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 2: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Note 3: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

System control status register 4

SYSSR4 (0x00FDF)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	(RVCTRS)	(RAREAS)	RSTDISS
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

Note 1: The enabled SYSCR3<RSTDIS> is initialized by a power-on reset only, and cannot be initialized by any other reset signals. The value written in SYSCR3 is reset by a power-on reset and other reset signals.

Note 2: Bits 7 to 3 of SYSCR4 are read as "0".

Downloaded from Arrow.com.

Internal factor reset detection status register

IRSTSR (0x00FCC)

	7	6	5	4	3	2	1	0
Bit Symbol	FCLR	-	-	-	-	LVDRF	SYSRF	WDTRF
Read/Write	W	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

FCLR	Flag initialization control	0 : 1 :	- Clears the internal factor reset flag to "0".
LVDRF	Voltage detection reset detection flag	0: 1:	- Detects the voltage detection reset.
SYSRF	System clock reset detection flag	0 : 1 :	- Detects the system clock reset.
WDTRF	Watchdog timer reset detection flag	0 : 1 :	- Detects the watchdog timer reset.

SYSRF which does not occur may be really set the same time.
Note that when checking the reset detection flag, check all bits of the reset detection flag to identify which internal factor resets occurred actually.

State of flag

	LVDRF	SYSRF	WDTRF
Detects the voltage detection reset.	1	* (Irregularity)	0
Detects the system clock reset.	0	1	0
Detects the watchdog timer reset.	0	* (Irregularity)	1

Note 1: Internal reset factor flag (IRSTSR<LVDRF, SYSRF, WDTRF>) is initialized only by a power-on reset, an external reset input or IRSTSR <FCLR>. It is not initialized by an internal factor reset.

Note 2: Care must be taken in system designing since the IRSTSR may not fulfill its functions due to disturbing noise and other effects.

Note 3: If SYSCR4 is set to 0x71 after IRSTSR<FCLR> is set to "1", internal factor reset flag is cleared to "0" and IRSTSR<FCLR> is automatically cleared to "0".

Note 4: After IRSTSR<FCLR> is modified, SYSCR4 should be written 0x71 (Enable code for IRSTSR<FCLR> in NORMAL mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

Note 5: Bit 7 to 3 of IRSTSR is read as "0".

2.4.3 Functions

The power-on reset, external reset input and internal factor reset signals are input to the warm-up circuit of the clock generator.

During reset, the warm-up counter circuit is reset, and the CPU and the peripheral circuits are reset.

After reset is released, the warm-up counter starts counting the high frequency clock (fc), and executes the warm-up operation that follows reset release.

When the warm-up operation and copying of the shadow RAM that follow reset release are finished, the CPU starts execution of the program from the reset vector address stored in addresses 0x1FFFE to 0x1FFFF.

When a reset signal is input during the warm-up operation and copying of the shadow RAM that follow reset release, the warm-up counter circuit is reset.

The reset operation is common to the power-on reset, external reset input and internal factor resets, except for the initialization of some special function registers and the initialization of the voltage detection circuits.

When a reset is applied, the peripheral circuits become the states as shown in Table 2-7.

Downloaded from Arrow.com.

Table 2-7 Initialization of Built-in Hardware by Reset Operation and Its Status after Release

Built-in hardware	During reset	During the warm-up operation that follows reset release	Immediately after the warm-up operation that follows reset release
Program counter (PC)	MCU mode: 0x1FFFE Serial PROM mode: 0x11FFE	MCU mode: 0x1FFFE Serial PROM mode: 0x11FFE	MCU mode: 0x1FFFE Serial PROM mode: 0x11FFE
Stack pointer (SP)	0x000FF	0x000FF	0x000FF
RAM	Indeterminate	Indeterminate	Indeterminate
General-purpose registers (W, A, B, C, D, E, H, L, IX and IY)	Indeterminate	Indeterminate	Indeterminate
Register bank selector (RBS)	0	0	0
Jump status flag (JF)	Indeterminate	Indeterminate	Indeterminate
Zero flag (ZF)	Indeterminate	Indeterminate	Indeterminate
Carry flag (CF)	Indeterminate	Indeterminate	Indeterminate
Half carry flag (HF)	Indeterminate	Indeterminate	Indeterminate
Sign flag (SF)	Indeterminate	Indeterminate	Indeterminate
Overflow flag (VF)	Indeterminate	Indeterminate	Indeterminate
Interrupt master enable flag (IMF)	0	0	0
Individual interrupt enable flag (EF)	0	0	0
Interrupt latch (IL)	0	0	0
Internal high-frequency clock oscillation circuit	Oscillation enabled	Oscillation enabled	Oscillation enabled
External high-frequency clock oscillation circuit	Oscillation disabled	Oscillation disabled	Oscillation disabled
External low-frequency clock oscillation circuit	Oscillation disabled	Oscillation disabled	Oscillation disabled
Warm-up counter	Reset	Start	Stop
Timing generator prescaler and divider	0	0	0
Watchdog timer	Disabled	Disabled	Enabled
Voltage detection circuit	Disabled or enabled	Disabled or enabled	Disabled or enabled
I/O port pin status	HiZ	HiZ	HiZ
Special function register	Refer to the SFR map.	Refer to the SFR map.	Refer to the SFR map.

Note: The voltage detection circuits are disabled by an external reset input or power-on reset only.

Downloaded from Arrow.com.

2.4.4 Reset Signal Generating Factors

Reset signals are generated by each factor as follows:

2.4.4.1 Power-on reset

The power-on reset is an internal reset that occurs when power is turned on.

During power-up, a power-on reset signal is generated while the supply voltage is below the power-on reset release voltage. When the supply voltage rises above the power-on reset release voltage, the power-on reset signal is released.

During power-down, a power-on reset signal is generated when the supply voltage falls below the power-on reset detection voltage.

Refer to "Power-on Reset circuit".

2.4.4.2 External reset input ($\overline{\text{RESET}}$ pin input)

This is an external reset that is generated by the $\overline{\text{RESET}}$ pin input. Port P10 is also used as the $\overline{\text{RESET}}$ pin, and it is configured as the $\overline{\text{RESET}}$ pin at power-up.

- During power-up
  - When the supply voltage rises rapidly

    When the power supply rise time (t_{VDD}) is shorter than 5 [ms] with enough margin, the reset can be released by a power-on reset or an external reset ($\overline{\text{RESET}}$ pin input).

    The power-on reset logic and external reset ($\overline{\text{RESET}}$ pin input) logic are ORed. This means that the TMP89FW24A is reset when either or both of these reset sources are asserted.

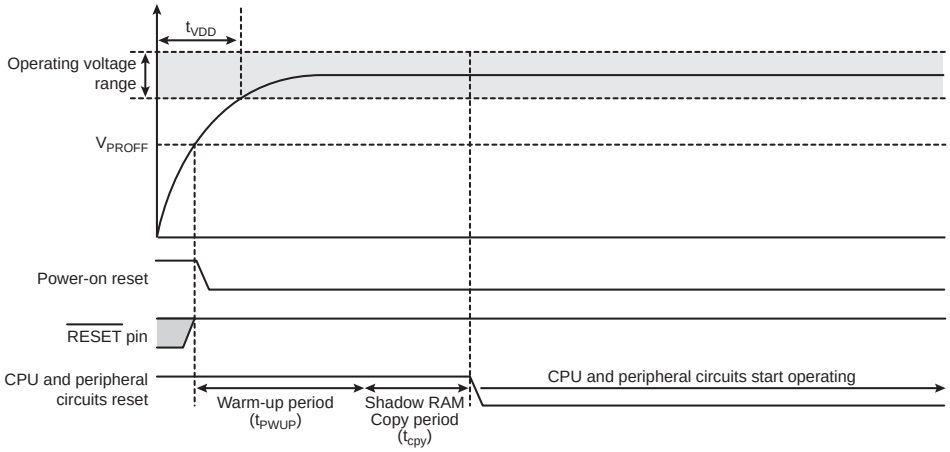
    Therefore, the reset time is determined by the reset source with a longer reset period. If the $\overline{\text{RESET}}$ pin level changes from Low to High before the supply voltage rises above the power-on-reset release voltage (V_{PROFF}) (or if the $\overline{\text{RESET}}$ pin level is High from the beginning), the reset time depends on the power-on reset. If the $\overline{\text{RESET}}$ pin level changes from Low to High after the supply voltage rises above V_{PROFF}, the reset time depends on the external reset.

    In the former case, a warm-up period begins when the power-on reset signal is released. In the latter case, a warm-up period begins when the $\overline{\text{RESET}}$ pin level becomes High. Upon completion of the warm-up period and copying of the shadow RAM, the CPU and peripheral circuits start operating (Figure 2-17).

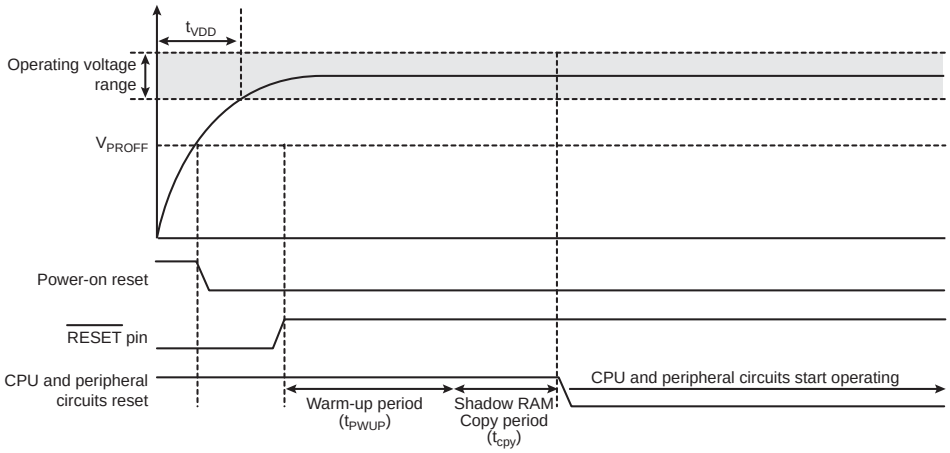
  - When the supply voltage rises slowly

    When the power supply rise time (t_{VDD}) is longer than 5 [ms], the reset must be released by using the $\overline{\text{RESET}}$ pin. In this case, hold the $\overline{\text{RESET}}$ pin Low until the supply voltage rises to the operating voltage range and oscillation is stabilized. When this state is achieved, wait at least 5 [μs] and then pull the $\overline{\text{RESET}}$ pin High. Changing the $\overline{\text{RESET}}$ pin level to High starts a warm-up period. Upon completion of the warm-up period and copying of the shadow RAM, the CPU and peripheral circuits start operating (Figure 2-17).

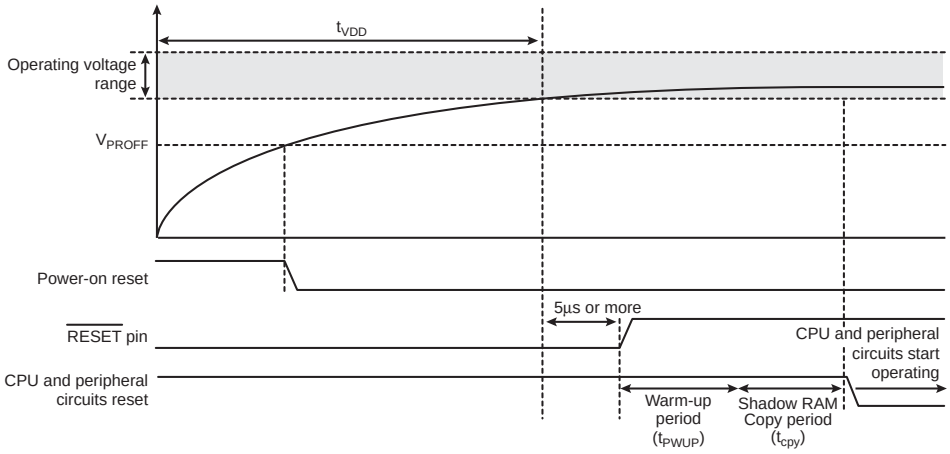
Downloaded from Arrow.com.



When the supply voltage rises rapidly (When the reset time depends on power-on reset)



When the supply voltage rises rapidly (When the reset time depends on external reset)



When the supply voltage rises slowly

Figure 2-17 External Reset Input (During Power-Up)

Downloaded from Arrow.com.

- When the supply voltage is within the operating voltage range

  When the supply voltage is within the operating voltage range and stable oscillation is achieved, holding the $\overline{\text{RESET}}$ pin Low for 5 [μs] or longer generates a reset. Then, changing the $\overline{\text{RESET}}$ pin level to High starts a warm-up period. Upon completion of the warm-up period and copying of the shadow RAM, the CPU and peripheral circuits start operating (Figure 2-18).

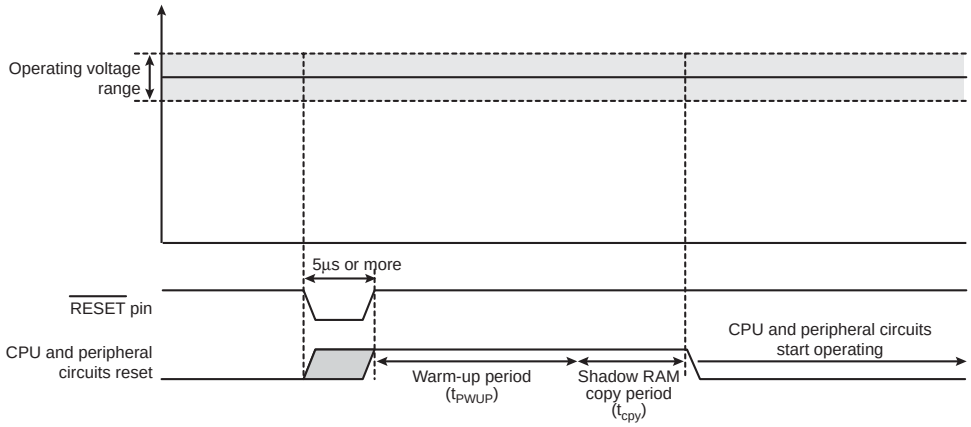


Figure 2-18 External Reset Input (When the Power Supply Is Stable)

2.4.4.3 Voltage detection reset

The voltage detection reset is an internal factor reset that occurs when it is detected that the supply voltage has reached a predetermined detection voltage.

Refer to "Voltage Detection Circuit".

2.4.4.4 Watchdog timer reset

The watchdog timer reset is an internal factor reset that occurs when an overflow of the watchdog timer is detected.

Refer to "Watchdog Timer".

2.4.4.5 System clock reset

The system clock reset is an internal factor reset that occurs when it is detected that the oscillation enable register is set to a combination that puts the CPU into deadlock.

Refer to "Clock Control Circuit".

2.4.4.6 Internal factor reset detection status register

By reading the internal factor reset detection status register IRSTSR after the release of an internal factor reset, except the power-on reset, the factor which causes a reset can be detected.

> SYSRF which does not occur may be really set the same time.
>
> Note that when checking the reset detection flag, check all bits of the reset detection flag to identify which internal factor resets occurred actually.

The internal factor reset detection status register is initialized by an external reset input or power-on reset.

Downloaded from Arrow.com.

Set IRSTSR<FCLR> to "1" and write 0x71 to SYSCR4. This enables IRSTSR<FCLR> and the internal factor reset detection status register is clear to "0". IRSTSR<FCLR> is cleared to "0" automatically after initializing the internal factor reset detection status register.

Note 1: Care must be taken in system designing since the IRSTSR may not fulfill its functions due to disturbing noise and other effects.

Note 2: When setting IRSTSR<FCLR> is to "1" and writing the enable code (0x71) to SYSCR4, these operations should be performed consecutively in NORMAL mode with the gear clock (fcgck) as fc/4 (OGCR<FCGCKSEL>=00) without switching to a different operating mode. Otherwise, IRSTSR<FCLR> may be enabled at unexpected timing.

2.4.4.7 How to use the external reset input pin as a port

To use the external reset input pin as a port, keep the external reset input pin at the "H" level until the power is turned on and the warm-up operation and copying of the shadow RAM that follow reset release are finished.

After the warm-up operation and copying of the shadow RAM that follow reset release are finished, set P1PU0 to "1" and P1CR0 to "0", and connect a pull-up resistor for a port. Then set SYSCR3<RSTDIS> to "1" and write 0xB2 to SYSCR4. This disables the external reset function and makes the external reset input pin usable as a normal port.

To use the pin as an external reset pin when it is used as a port, set P1PU0 to "1" and P1CR0 to "0" and connect the pull-up resistor to put the pin to the input mode. Then clear SYSCR3<RSTDIS> to "0" and write 0xB2 to SYSCR4. This enables the external reset function and makes the pin usable as the external reset input pin.

Note 1: If you switch the external reset input pin to a port or switch the pin used as a port to the external reset input pin, do it when the pin is stabilized at the "H" level. Switching the pin function when the "L" level is input may cause a reset.

Note 2: If the external reset input is used as a port, the statement which clears SYSCR3<RSTDIS> to "0" is not written in a program. By the abnormal execution of program, the external reset input set as a port may be changed as the external reset input at unexpected timing.

Note 3: After SYSCR3<RSTDIS> is modified, SYSCR4 should be written 0xB2 (Enable code for SYSCR3<RSTDIS>) in NORMAL1 mode when fcgck is fc/4 (CGCR<FCGCKSEL>=00). Otherwise, SYSCR3<RSTDIS> may be enabled at unexpected timing.

Downloaded from Arrow.com.

3. Interrupt Control Circuit

The TMP89FW24A has a total of 30 interrupt sources excluding reset. Interrupts can be nested with priorities. Three of the internal interrupt sources are non-maskable while the rest are maskable.

Interrupt sources are provided with interrupt latches (IL), which hold interrupt requests, and have independent vector addresses. When a request for an interrupt is generated, its interrupt latch is set to "1", which requests the CPU to accept the interrupt. Acceptance of interrupts is enabled or disabled by software using the interrupt master enable flag (IMF) and individual enable flag (EF) for each interrupt source. If multiple maskable interrupts are generated simultaneously, the interrupts are accepted in order of descending priority. The priorities are determined by the interrupt priority change control register (ILPRS1-ILPRS7) as Levels and determined by the hardware as the basic priorities.

However, there are no prioritized interrupt sources among non-maskable interrupts.

Interrupt sources		Enable condition	Interrupt latch	Vector Address (MCU mode)		Basic priority
				RVCTR=0 enabled	RVCTR=1 enabled	
Internal/External	(Reset)	Non-maskable	-	0xFFFE	-	1
Internal	INTSWI	Non-maskable	-	0xFFFC	0x01FC	2
Internal	INTUNDEF	Non-maskable	-	0xFFFC	0x01FC	2
Internal	INTWDT	Non-maskable	ILL<IL3>	0xFFF8	0x01F8	2
Internal	INTWUC	IMF AND EIRL<EF4> = 1	ILL<IL4>	0xFFF6	0x01F6	5
Internal	INTTBT	IMF AND EIRL<EF5> = 1	ILL<IL5>	0xFFF4	0x01F4	6
Internal	INTRXD0	IMF AND EIRL<EF6> = 1	ILL<IL6>	0xFFF2	0x01F2	7
Internal	INTTXD0	IMF AND EIRL<EF7> = 1	ILL<IL7>	0xFFF0	0x01F0	8
External	$\overline{\text{INT5}}$	IMF AND EIRH<EF8> = 1	ILH<IL8>	0xFFEE	0x01EE	9
Internal	INTVLTD	IMF AND EIRH<EF9> = 1	ILH<IL9>	0xFFEC	0x01EC	10
Internal	INTADC	IMF AND EIRH<EF10> = 1	ILH<IL10>	0xFFEA	0x01EA	11
Internal	INTRTC	IMF AND EIRH<EF11> = 1	ILH<IL11>	0xFFE8	0x01E8	12
Internal	INTTC00	IMF AND EIRH<EF12> = 1	ILH<IL12>	0xFFE6	0x01E6	13
Internal	INTTC01	IMF AND EIRH<EF13> = 1	ILH<IL13>	0xFFE4	0x01E4	14
Internal	INTTCA0	IMF AND EIRH<EF14> = 1	ILH<IL14>	0xFFE2	0x01E2	15
Internal	INTSIO0	IMF AND EIRH<EF15> = 1	ILH<IL15>	0xFFE0	0x01E0	16
External	INT0	IMF AND EIRE<EF16> = 1	ILE<IL16>	0xFFDE	0x01DE	17
External	INT1	IMF AND EIRE<EF17> = 1	ILE<IL17>	0xFFDC	0x01DC	18
External	INT2	IMF AND EIRE<EF18> = 1	ILE<IL18>	0xFFDA	0x01DA	19
External	INT3	IMF AND EIRE<EF19> = 1	ILE<IL19>	0xFFD8	0x01D8	20
Internal	INTEMG0	IMF AND EIRE<EF20> = 1	ILE<IL20>	0xFFD6	0x01D6	21
Internal	INTTCB0	IMF AND EIRE<EF21> = 1	ILE<IL21>	0xFFD4	0x01D4	22
Internal	INTRXD1	IMF AND EIRE<EF22> = 1	ILE<IL22>	0xFFD2	0x01D2	23
Internal	INTTXD1	IMF AND EIRE<EF23> = 1	ILE<IL23>	0xFFD0	0x01D0	24
Internal	INTTC02	IMF AND EIRD<EF24> = 1	ILD<IL24>	0xFFCE	0x01CE	25
Internal	INTTC03	IMF AND EIRD<EF25> = 1	ILD<IL25>	0xFFCC	0x01CC	26
Internal	INTRXD2	IMF AND EIRD<EF26> = 1	ILD<IL26>	0xFFCA	0x01CA	27
Internal	INTTXD2	IMF AND EIRD<EF27> = 1	ILD<IL27>	0xFFC8	0x01C8	28
Internal	INTSIO1/INTSBI0	IMF AND EIRD<EF28> = 1	ILD<IL28>	0xFFC6	0x01C6	29
Internal	INTTCC0P	IMF AND EIRD<EF29> = 1	ILD<IL29>	0xFFC4	0x01C4	30
Internal	INTTCC0T	IMF AND EIRD<EF30> = 1	ILD<IL30>	0xFFC2	0x01C2	31
-	-	-	-	-	-	-

Downloaded from Arrow.com.

Note 1: Vector address areas can be changed by the SYSCR3<RVCTR> setting. To assign vector address areas to RAM, set SYSCR3<RVCTR> to "1" and SYSCR3<RAREA> to "1".

Note 2: 0x1FFFA and 0x1FFFB function not as interrupt vectors but as option codes in the serial PROM mode. For details, see "Serial PROM Mode".

Note 3: Do not set SYSCR3<RVCTR> to "0" in the serial PROM mode. If an interrupt is generated with SYSCR3<RVCTR> ="0", the software refers to the vector area in the BOOTROM and the user cannot use it.

2026-04-21 dl#
Downloaded from Arrow.com.

3.1 Configuration

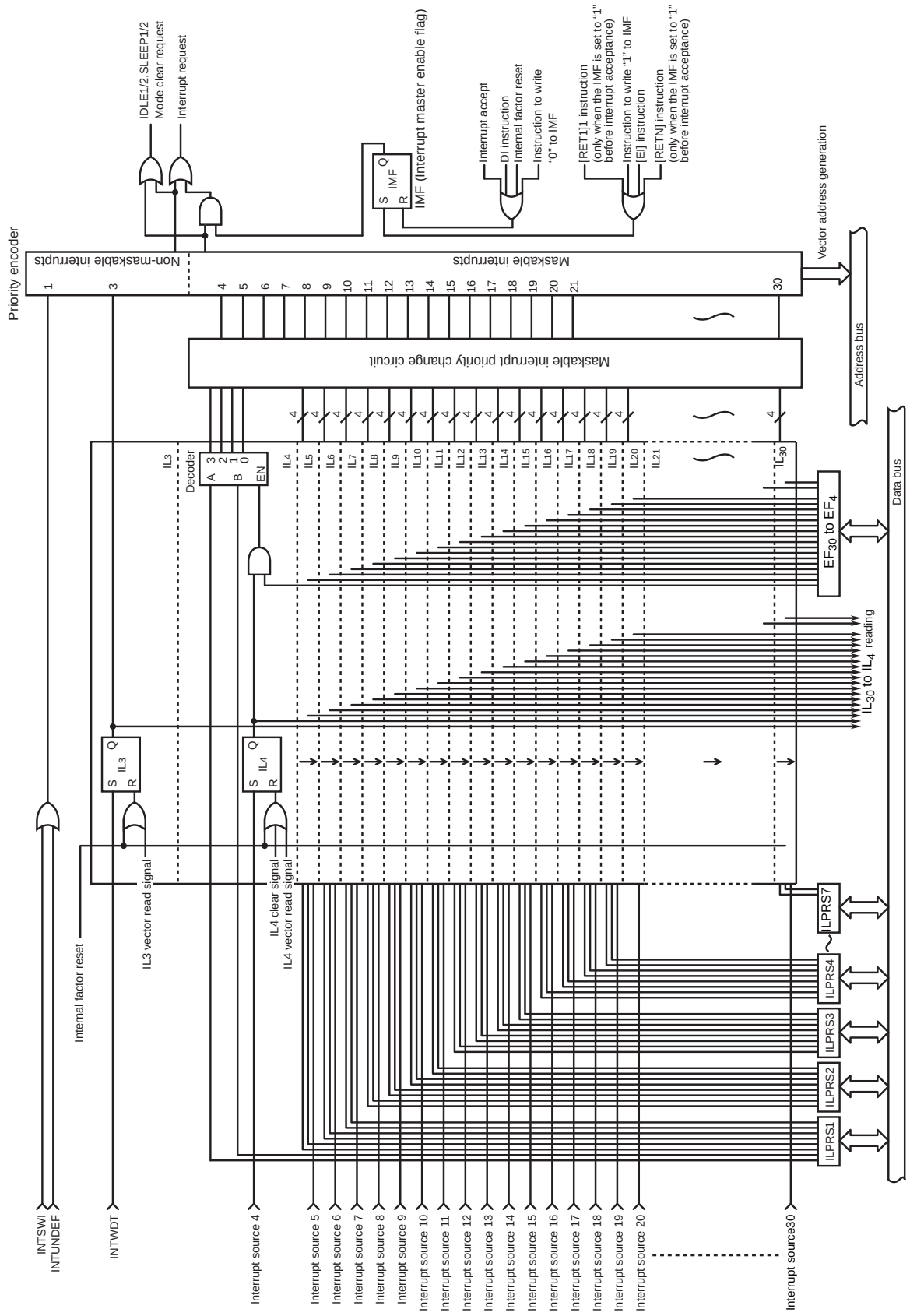


Figure 3-1 Interrupt Control Circuit

Downloaded from Arrow.com.

3.2 Interrupt Latches (IL30 to IL3)

An interrupt latch is provided for each interrupt source, except for a software interrupt and an undefined instruction execution interrupt. When an interrupt request is generated, the latch is set to "1", and the CPU is requested to accept the interrupt if its acceptance is enabled. The interrupt latch is cleared to "0" immediately after the interrupt is accepted. All interrupt latches are initialized to "0" during reset.

The interrupt latches are located at addresses 0x0FE0, 0x0FE1, 0x0FE2, 0x0FE3 in SFR area. Each latch can be cleared to "0" individually by an instruction. However, IL2 and IL3 interrupt latches cannot be cleared by instructions.

Do not use any read-modify-write instruction, such as a bit manipulation or operation instruction, because it may clear interrupt requests generated while the instruction is executed.

Interrupt latches cannot be set to "1" by using an instruction. Writing "1" to an interrupt latch is equivalent to denying clearing of the interrupt latch, and not setting the interrupt latch.

Since interrupt latches can be read by instructions, the status of interrupt requests can be monitored by software.

Note: In the main program, before manipulating an interrupt latch (IL), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the IL (Enable interrupt by EI instruction).

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the IL before setting the IMF to "1".

Example 1:Clears interrupt latches

```
DI                                  ;IMF ← 0
LD      (ILL), 0y00111111           ;IL7 to IL6 ← 0
LD      (ILH), 0y11101000           ;IL12, IL10 to IL8 ← 0
EI                                  ;IMF ← 1
```

Example 2:Reads interrupt latches

```
LD      WA, (ILL)                   ;W ← ILH, A ← ILL
```

Example 3:Tests interrupt latches

```
TEST    (ILL). 7                    ;if IL7=1 then jump
JR      F, code_addr(SSET)          ;
```


2026-04-21
Downloaded from Arrow.com.

3.3 Interrupt Enable Register (EIR)

The interrupt enable register (EIR) enables and disables the acceptance of interrupts, except for the non-maskable interrupts (software interrupt, undefined instruction interrupt and watchdog interrupt). Non-maskable interrupts are accepted regardless of the contents of the EIR.

The EIR consists of the interrupt master enable flag (IMF) and the individual interrupt enable flags (EF). These registers are located at addresses 0x003A, 0x003B, 0x003C, 0x003D in the SFR area, and they can be read and written by instructions (including read-modify-write instructions such as bit manipulation or operation instructions).

3.3.1 Interrupt master enable flag (IMF)

The interrupt master enable flag (IMF) enables and disables the acceptance of all maskable interrupts. Clearing the IMF to "0" disables the acceptance of all maskable interrupts. Setting the IMF to "1" enables the acceptance of the interrupts that are specified by the individual interrupt enable flags.

When an interrupt is accepted, the IMF is stacked and then cleared to "0", which temporarily disables the subsequent maskable interrupts. After the interrupt service routine is executed, the stacked data, which was the status before interrupt acceptance, reloaded on the IMF by return interrupt instruction [RETI]/[RETN].

The IMF is located on bit 0 in EIRL (Address: 0x03A in SFR), and can be read and written by instructions. The IMF is normally set and cleared by [EI] and [DI] instructions respectively. During reset, the IMF is initialized to "0".

3.3.2 Individual interrupt enable flags (EF30 to EF4)

Each of these flags enables and disables the acceptance of its maskable interrupt. Setting the corresponding bit of an individual interrupt enable flag to "1" enables acceptance of its interrupt, and setting the bit to "0" disables acceptance.

During reset, all the individual interrupt enable flags are initialized to "0" and no maskable interrupts are accepted until the flags are set to "1".

Note: In the main program, before manipulating the interrupt enable flag (EF), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the EF (Enable interrupt by EI instruction).

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the EF before setting the IMF to "1".

Example: Enables interrupts individually and sets IMF

```
DI                                      ;IMF ← 0
LDW     (EIRL), 0y1110100010100000      ;EF15 to EF13, EF11, EF7, EF5 ← 1
:                                       ;Note: IMF should not be set.
:
EI                                      ;IMF ← 1
```


#!Undefined!#
Downloaded from Arrow.com.

Interrupt latch (ILL)

ILL (0x00FE0)		7	6	5	4	3	2	1	0
	Bit Symbol	IL7	IL6	IL5	IL4	IL3	-	-	-
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD0	INTRXD0	INTTBT	INTWUC	INTWDT			

Interrupt latch (ILH)

ILH (0x00FE1)		7	6	5	4	3	2	1	0
	Bit Symbol	IL15	IL14	IL13	IL12	IL11	IL10	IL9	IL8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTSIO0	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTVLTD	INT5

Interrupt latch (ILE)

ILE (0x00FE2)		7	6	5	4	3	2	1	0
	Bit Symbol	IL23	IL22	IL21	IL20	IL19	IL18	IL17	IL16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD1	INTRXD1	INTTCB0	INTEMG0	INT3	INT2	INT1	INT0

Interrupt latch (ILD)

ILD (0x00FE3)		7	6	5	4	3	2	1	0
	Bit Symbol	-	IL30	IL29	IL28	IL27	IL26	IL25	IL24
	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function		INTTCC0T	INTTCC0P	INTSIO1/ INTSBI0	INTTXD2	INTRXD2	INTTC03	INTTC02

			Read	Write
IL30 to IL4	Interrupt latch	0:	No interrupt request	Clears the interrupt request (Notes 2 and 3)
		1:	Interrupt request	Does not clear the interrupt request (Interrupt is not set by writing "1".)
IL3		0: 1:	No interrupt request Interrupt request	-

Note 1: IL3 is a read-only register. Writing the register does not affect interrupt latch.

Note 2: In the main program, before manipulating an interrupt latch (IL), be sure to clear the interrupt master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the IL (Enable interrupt by EI instruction).

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the IL before setting the IMF to "1".

Note 3: Do not clear IL with read-modify-write instructions such as bit operations.

Note 4: When a read instruction is executed on ILL, bits 0 to 2 are read as "0". Other unused bits are read as "0".

2026-04-21
Downloaded from Arrow.com.

Interrupt enable register (EIRL)

EIRL (0x0003A)		7	6	5	4	3	2	1	0
	Bit Symbol	EF7	EF6	EF5	EF4	-	-	-	IMF
	Read/Write	R/W	R/W	R/W	R/W	R	R	R	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD0	INTRXD0	INTTBT	INTWUC				Interrupt master enable flag

Interrupt enable register (EIRH)

EIRH (0x0003B)		7	6	5	4	3	2	1	0
	Bit Symbol	EF15	EF14	EF13	EF12	EF11	EF10	EF9	EF8
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTSIO0	INTTCA0	INTTC01	INTTC00	INTRTC	INTADC	INTVLTD	INT5

Interrupt enable register (EIRE)

EIRE (0x0003C)		7	6	5	4	3	2	1	0
	Bit Symbol	EF23	EF22	EF21	EF20	EF19	EF18	EF17	EF16
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	INTTXD1	INTRXD1	INTTCB0	INTEMG0	INT3	INT2	INT1	INT0

Interrupt enable register (EIRD)

EIRD (0x0003D)		7	6	5	4	3	2	1	0
	Bit Symbol	-	EF30	EF29	EF28	EF27	EF26	EF25	EF24
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function		INTTCC0T	INTTCC0P	INTSIO1/ INTSBI0	INTTXD2	INTRXD2	INTTC03	INTTC02

EF30 to EF4	Individual interrupt enable flag (Specified for each bit)	0: 1:	Disables the acceptance of each maskable interrupt. Enables the acceptance of each maskable interrupt.
IMF	Interrupt master enable flag	0: 1:	Disables the acceptance of all maskable interrupts. Enables the acceptance of all maskable interrupts.

Note 1: Do not set the IMF and the interrupt enable flag (EF15 to EF4) to "1" at the same time.

Note 2: In the main program, before manipulating the interrupt enable flag (EF), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction). Then set the IMF to "1" as required after operating the EF (Enable interrupt by EI instruction)

In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate the EF before setting the IMF to "1".

Note 3: When a read instruction is executed on EIRL, bits 3 to 1 are read as "0". Other unused bits are read as "0".

Downloaded from Arrow.com.

3.4 Maskable Interrupt Priority Change Function

The priority of maskable interrupts (IL4 to IL30) can be changed to four levels, Levels 0 to 3, regardless of the basic priorities 5 to 31. Interrupt priorities can be changed by the interrupt priority change control register (ILPRS1 to ILPRS7). To raise the interrupt priority, set the Level to a larger number. To lower the interrupt priority, set the Level to a smaller number. When different maskable interrupts are generated simultaneously at the same level, the interrupt with higher basic priority is processed preferentially. For example, when the ILPRS1 register is set to 0xC0 and interrupts IL4 and IL7 are generated at the same time, IL7 is preferentially processed (provided that EF4 and EF7 have been enabled).

After reset is released, all maskable interrupts are set to priority level 0 (the lowest priority).

Note: In the main program, before manipulating the interrupt priority change control register (ILPRS1 to 7), be sure to clear the master enable flag (IMF) to "0" (Disable interrupt by DI instruction).
Set the IMF to "1" as required after operating ILPRS1 to 7 (Enable interrupt by EI instruction).
In the interrupt service routine, the IMF becomes "0" automatically and need not be cleared to "0" normally. However, if using multiple interrupt in the interrupt service routine, manipulate ILPRS1 to 7 before setting the IMF to "1".

Interrupt priority change control register 1

ILPRS1 (0x00FF0)	7	6	5	4	3	2	1	0
Bit Symbol	IL07P		IL06P		IL05P		IL04P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL07P	Sets the interrupt priority of IL7.	00:	Level 0 (lower priority)
IL06P	Sets the interrupt priority of IL6.	01:	Level 1
IL05P	Sets the interrupt priority of IL5.	10:	Level 2
IL04P	Sets the interrupt priority of IL4.	11:	Level 3 (higher priority)

Interrupt priority change control register 2

ILPRS2 (0x00FF1)	7	6	5	4	3	2	1	0
Bit Symbol	IL11P		IL10P		IL09P		IL08P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL11P	Sets the interrupt priority of IL11.	00:	Level 0 (lower priority)
IL10P	Sets the interrupt priority of IL10.	01:	Level 1
IL09P	Sets the interrupt priority of IL9.	10:	Level 2
IL08P	Sets the interrupt priority of IL8.	11:	Level 3 (higher priority)

Interrupt priority change control register 3

ILPRS3 (0x00FF2)	7	6	5	4	3	2	1	0
Bit Symbol	IL15P		IL14P		IL13P		IL12P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL15P	Sets the interrupt priority of IL15.	00:	Level 0 (lower priority)
IL14P	Sets the interrupt priority of IL14.	01:	Level 1
IL13P	Sets the interrupt priority of IL13.	10:	Level 2
IL12P	Sets the interrupt priority of IL12.	11:	Level 3 (higher priority)

Downloaded from Arrow.com.

Interrupt priority change control register 4

ILPRS4 (0x00FF3)	7	6	5	4	3	2	1	0
Bit Symbol	IL19P		IL18P		IL17P		IL16P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL19P	Sets the interrupt priority of IL19.	00:	Level 0 (lower priority)
IL18P	Sets the interrupt priority of IL18.	01:	Level 1
IL17P	Sets the interrupt priority of IL17.	10:	Level 2
IL16P	Sets the interrupt priority of IL16.	11:	Level 3 (higher priority)

Interrupt priority change control register 5

ILPRS5 (0x00FF4)	7	6	5	4	3	2	1	0
Bit Symbol	IL23P		IL22P		IL21P		IL20P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL23P	Sets the interrupt priority of IL23.	00:	Level 0 (lower priority)
IL22P	Sets the interrupt priority of IL22.	01:	Level 1
IL21P	Sets the interrupt priority of IL21.	10:	Level 2
IL20P	Sets the interrupt priority of IL20.	11:	Level 3 (higher priority)

Interrupt priority change control register 6

ILPRS6 (0x00FF5)	7	6	5	4	3	2	1	0
Bit Symbol	IL27P		IL26P		IL25P		IL24P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

IL27P	Sets the interrupt priority of IL27.	00:	Level 0 (lower priority)
IL26P	Sets the interrupt priority of IL26.	01:	Level 1
IL25P	Sets the interrupt priority of IL25.	10:	Level 2
IL24P	Sets the interrupt priority of IL24.	11:	Level 3 (higher priority)

Interrupt priority change control register 7

ILPRS7 (0x00FF6)	7	6	5	4	3	2	1	0
Bit Symbol			IL30P		IL29P		IL28P	
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0

		00:	Level 0 (lower priority)
IL30P	Sets the interrupt priority of IL30.	01:	Level 1
IL29P	Sets the interrupt priority of IL29.	10:	Level 2
IL28P	Sets the interrupt priority of IL28.	11:	Level 3 (higher priority)

Downloaded from Arrow.com.

3.5 Interrupt Sequence

An interrupt request, which raised interrupt latch, is held, until interrupt is accepted or interrupt latch is cleared to “0” by resetting or an instruction. Interrupt acceptance sequence requires 8-machine cycles after the completion of the current instruction. The interrupt service task terminates upon execution of an interrupt return instruction [RETI] (for maskable interrupts) or [RETN] (for non-maskable interrupts).

3.5.1 Initial Setting

Using an interrupt requires specifying an SP (stack pointer) for it in advance. The SP is a 16-bit register pointing at the start address of a stack. The SP is post-decremented when a subroutine call or a push instruction is executed or when an interrupt request is accepted. It is pre-incremented when a return or pop instruction is executed. Therefore, the stack becomes deeper toward lower stack location addresses. Be sure to reserve a stack area having an appropriate size based on the SP setting.

The SP is initialized to 00FFH after a reset. If you need to change the SP, do so right after a reset or when the interrupt master enable flag (IMF) is “0”.

Example :SP setting

```
LD      SP, 023FH     ; SP = 023FH
LD      SP, SP+04H    ; SP = SP + 04H
ADD     SP, 0010H     ; SP = SP + 0010H
```

3.5.2 Interrupt acceptance processing

Interrupt acceptance processing is packaged as follows.

1. The interrupt master enable flag (IMF) is cleared to “0” in order to disable the acceptance of any following interrupt.
2. The interrupt latch (IL) for the interrupt source accepted is cleared to “0”.
3. The contents of the program counter (PC) and the program status word, including the interrupt master enable flag (IMF), are saved (Pushed) on the stack in sequence of PSW + IMF, PCH, PCL. Meanwhile, the stack pointer (SP) is decremented by 3.
4. The entry address (Interrupt vector) of the corresponding interrupt service program, loaded on the vector table, is transferred to the program counter.
5. The instruction stored at the entry address of the interrupt service program is executed.

Note: When the contents of PSW are saved on the stack, the contents of register bank and IMF are also saved.

Example: Correspondence between vector table address for INTTBT and the entry address of the interrupt service program

2026-04-21
Downloaded from Arrow.com.

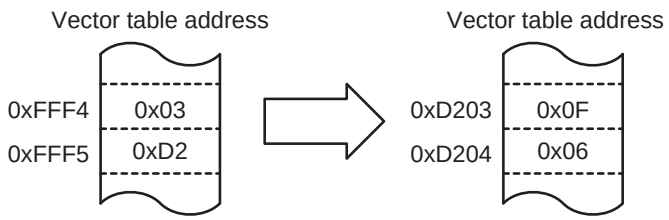


Figure 3-2 Vector table address and Entry address

Downloaded from Arrow.com.

A maskable interrupt is not accepted until the IMF is set to “1” even if the maskable interrupt is requested in the interrupt service routine.

In order to utilize nested interrupt service, the IMF must be set to “1” in the interrupt service program. In this case, acceptable interrupt sources are selectively enabled by the individual interrupt enable flags.

To avoid overloaded nesting, clear the individual interrupt enable flag whose interrupt is currently serviced, before setting IMF to “1”. As for non-maskable interrupt, keep interrupt service shorter compared with length between interrupt requests.

3.5.3 Saving/restoring general-purpose registers

During interrupt acceptance processing, the program counter (PC) and the program status word (PSW, includes IMF) are automatically saved on the stack, but the general purpose registers are not. These registers must be saved by software if necessary. When multiple interrupt services are nested, it is also necessary to avoid using the same data memory area for saving registers. The following methods are used to save/restore the general-purpose registers.

3.5.3.1 Using PUSH and POP instructions

To save only a specific register, PUSH and POP instructions are available.

Example :Using PUSH and POP instructions

```
PINTxx    PUSH                WA    ; Save WA register
          Interrupt processing
          POP                 WA    ; Restore WA register
          RETI                      ; RETURN
```

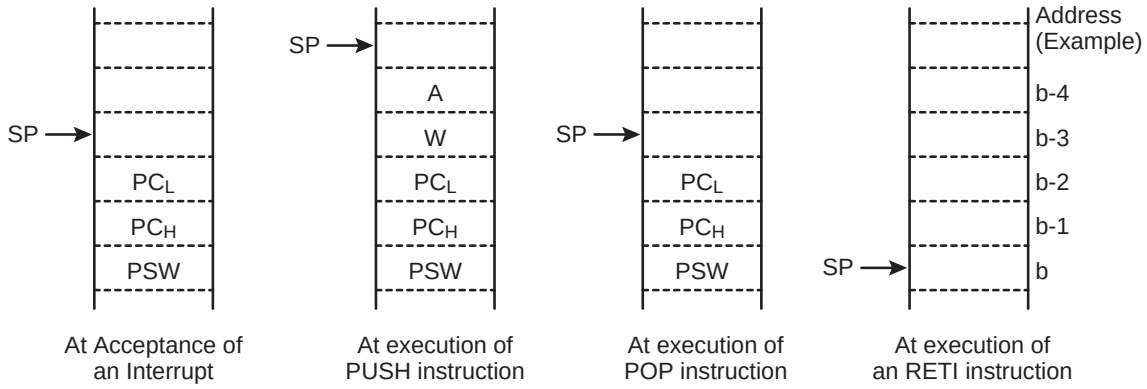


Figure 3-3 Saving/restoring general-purpose registers

2026-04-21
Downloaded from Arrow.com.

3.5.3.2 Using data transfer instructions

To save only a specific register without nested interrupts, data transfer instructions are available.

Example :Save/store register using data transfer instructions

```
PINTxx:   LD                (GSAVA), A      ; Save A register
          Interrupt processing
          LD                A, (GSAVA)      ; Restore A register
          RETI                              ; RETURN
```

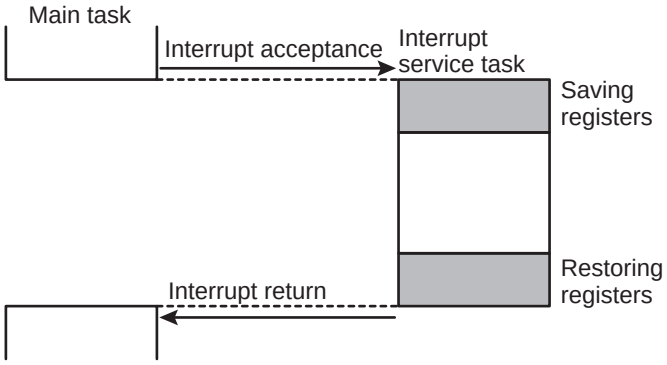


Figure 3-4 Saving/Restoring General-purpose Registers under Interrupt Processing

3.5.3.3 Using a register bank to save/restore general-purpose registers

In non-multiple interrupt handling, the register bank function can be used to save/restore the general-purpose registers at a time. The register bank function saves (switches) the general-purpose registers by executing a register bank manipulation instruction (such as LD RBS,1) at the beginning of an interrupt service task. It is unnecessary to re-execute the register bank manipulation instruction at the end of the interrupt service task because executing the RETI instruction makes a return automatically to the register bank that was being used by the main task according to the content of the PSW.

Note: Two register banks (BANK0 and BANK1) are available. Each bank consists of 8-bit general-purpose registers (W, A, B, C, D, E, H, and L) and 16-bit general-purpose registers (IX and IY).

Example :Saving/restoring registers, using an instruction for transfer with data memory (with the main task using the register bank BANK0)

```
PINTxx:   LD                RBS, 1          ; Switches to the register bank BANK1
          Interrupt processing
          RETI                              ; RETURN
                                            (Makes a return automatically to BANK0
                                            that was being used by the main task when
                                            the PSW is restored)
```


Downloaded from Arrow.com.

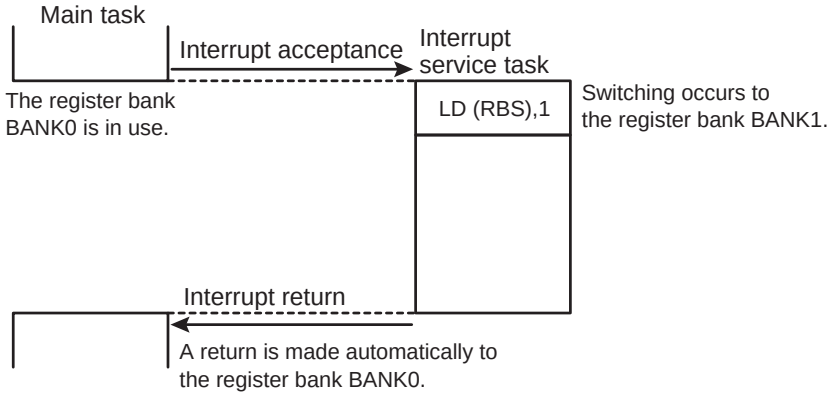


Figure 3-5 Saving/Restoring General-purpose Registers under Interrupt Processing

3.5.4 Interrupt return

Interrupt return instructions [RETI]/[RETN] perform as follows.

[RETI]/[RETN] Interrupt Return
1. Program counter (PC) and program status word (register bank) are restored from tha stack. 2. Stack pointer (SP) is incremented by 3.

Downloaded from Arrow.com.

3.6 Software Interrupt (INTSW)

Executing the SWI instruction generates a software interrupt and immediately starts interrupt processing (INTSW is the top-priority interrupt).

Use the SWI instruction only for address error detection or for debugging described below.

3.6.1 Address error detection

0xFF is read if for some cause such as noise the CPU attempts to fetch an instruction from a non-existent memory address. Code 0xFF is an SWI instruction, so a software interrupt is generated and an address error is detected. The address error detection range can be further expanded by writing 0xFF to unused areas in the program memory.

3.6.2 Debugging

Debugging efficiency can be increased by placing the SWI instruction at the software break point setting address.

3.7 Undefined Instruction Interrupt (INTUNDEF)

When the CPU tries to fetch and execute an instruction that is not defined, INTUNDEF is generated and starts the interrupt processing. INTUNDEF is accepted even if another non-maskable interrupt is in process. The current process is discontinued and the INTUNDEF interrupt process starts soon after it is requested.

Note: The undefined instruction interrupt (INTUNDEF) forces the CPU to jump into the interrupt vector address, as software interrupt (SWI) does.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

4. External Interrupt control circuit

External interrupts detects the change of the input signal and generates an interrupt request. Noise can be removed by the built-in digital noise canceller.

4.1 Configuration

The external interrupt control circuit consists of a noise canceller, an edge detection circuit, a level detection circuit and an interrupt signal generation circuit.

Externally input signals are input to the rising edge or falling edge or level detection circuit for each external interrupt, after noise is removed by the noise canceller.

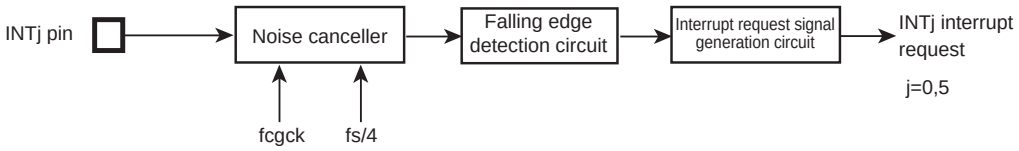


Figure 4-1 External Interrupts 0/5

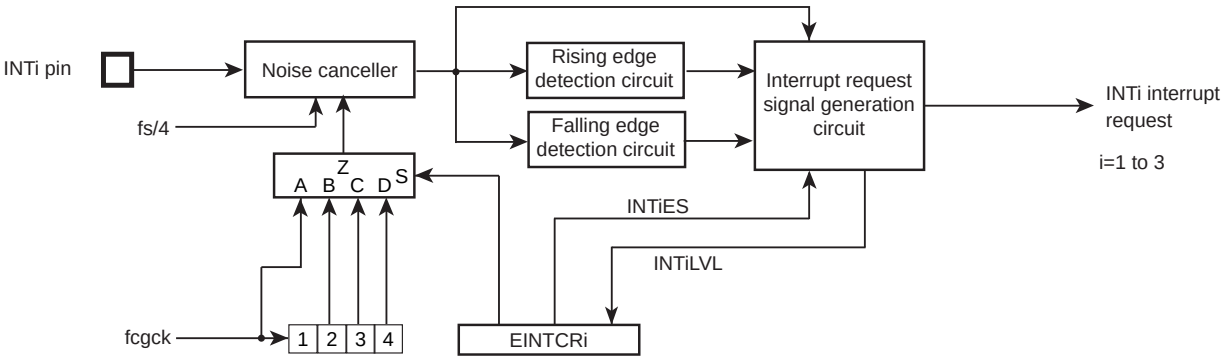


Figure 4-2 External Interrupts 1/2/3

4.2 Control

External interrupts are controlled by the following registers:

Downloaded from Arrow.com.

Low power consumption register 3

POFFCR3 (0x00F77)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	INT5EN	-	INT3EN	INT2EN	INT1EN	INT0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 Control	0 1	Disable Enable
INT3EN	INT3 Control	0 1	Disable Enable
INT2EN	INT2 Control	0 1	Disable Enable
INT1EN	INT1 Control	0 1	Disable Enable
INT0EN	INT0 Control	0 1	Disable Enable

Note 1: Clearing INTxEN(x=0 to 5) to "0" stops the clock supply to the external interrupts. This invalidates the data written in the control register for each external interrupt. When using the external interrupts, set INTxEN to "1" and then write data into the control register for each external interrupt.

Note 2: Interrupt request signals may be generated when INTxEN is changed. Before changing INTxEN, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Bits 7 and 6 of POFFSET3 are read as "0".

External interrupt control register 1

EINTCR1 (0x00FD8)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT1LVL	INT1ES		INT1NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INI1LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 1	0 : 1 :	Initial state or signal level "L" Signal level "H"		
INT1ES	Selects the interrupt request generating condition for external interrupt 1	00 : 01 : 10 : 11 :	An interrupt request is generated at the rising edge of the noise canceller pass signal An interrupt request is generated at the falling edge of the noise canceller pass signal An interrupt request is generated at both edges of the noise canceller pass signal Reserved		
INT1NC	Sets the noise canceller sampling interval for external interrupt 1		NORMAL1/2, IDLE1/2		SLOW1/2, SLEEP1
		00 : 01 : 10 : 11 :	fcgck [Hz] fcgck / 2^2 [Hz] fcgck / 2^3 [Hz] fcgck / 2^4 [Hz]	00 : 01 : 10 : 11 :	fs/4 [Hz] fs/4 [Hz] fs/4 [Hz] fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR1 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NOR-

Downloaded from Arrow.com.

MAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR1 are read as "0".

External interrupt control register 2

EINTCR1 (0x00FD9)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT2LVL	INT2ES		INT2NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INI2LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 2	0 :	Initial state or signal level "L"		
		1 :	Signal level "H"		
INT2ES	Selects the interrupt request generating condition for external interrupt 2	00 :	An interrupt request is generated at the rising edge of the noise canceller pass signal		
		01 :	An interrupt request is generated at the falling edge of the noise canceller pass signal		
		10 :	An interrupt request is generated at both edges of the noise canceller pass signal		
		11 :	Reserved		
INT2NC	Sets the noise canceller sampling interval for external interrupt 2		NORMAL1/2, IDLE1/2		SLOW1/2, SLEEP1
		00 :	fcgck [Hz]	00 :	fs/4 [Hz]
		01 :	fcgck / 2^2 [Hz]	01 :	fs/4 [Hz]
		10 :	fcgck / 2^3 [Hz]	10 :	fs/4 [Hz]
		11 :	fcgck / 2^4 [Hz]	11 :	fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR2 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR2 are read as "0".

Downloaded from Arrow.com.

External interrupt control register 3

EINTCR3 (0x00FDA)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	INT3LVL	INT3ES		INT3NC	
Read/Write	R	R	R	R	R/W		R/W	
After reset	0	0	0	0	0		0	

INI3LVL	Noise canceller pass signal level when the interrupt request signal is generated for external interrupt 3	0 :	Initial state or signal level "L"		
		1 :	Signal level "H"		
INT3ES	Selects the interrupt request generating condition for external interrupt 3	00 :	An interrupt request is generated at the rising edge of the noise canceller pass signal		
		01 :	An interrupt request is generated at the falling edge of the noise canceller pass signal		
		10 :	An interrupt request is generated at both edges of the noise canceller pass signal		
		11 :	Reserved		
INT3NC	Sets the noise canceller sampling interval for external interrupt 3		NORMAL1/2, IDLE1/2		SLOW1/2, SLEEP1
		00 :	fcgck [Hz]	00 :	fs/4 [Hz]
		01 :	fcgck / 2^2 [Hz]	01 :	fs/4 [Hz]
		10 :	fcgck / 2^3 [Hz]	10 :	fs/4 [Hz]
		11 :	fcgck / 2^4 [Hz]	11 :	fs/4 [Hz]

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 3: Interrupt requests may be generated when EINTCR3 is changed. Before doing such operation, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

Note 4: Bits 7 to 5 of EINTCR3 are read as "0".

4.3 Function

The condition for generating interrupt request signals and the noise cancel time can be set for external interrupts 1 to 3.

The condition for generating interrupt request signals and the noise cancel time are fixed for external interrupts 0 and 5.

Downloaded from Arrow.com.

Table 4-1 External Interrupts

Source	Pin	Enable conditions	Interrupt request signal generated at	External interrupt pin input signal width and noise removal	
				NORMAL1/2, IDLE1/2	SLOW1/2, SLEEP1
INT0	$\overline{INT0}$	IMF AND EF16 = 1	Falling edge	Less than 1/fcgck: Noise More than 1/fcgck and less than 2/fcgck: Indeterminate More than 2/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT1	INT1	IMF AND EF17 = 1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT2	INT2	IMF AND EF18 = 1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT3	INT3	IMF AND EF19 = 1	Falling edge Rising edge Both edges	Less than 2/fspl: Noise More than 2/fspl and less than 3/fspl+1/fcgck: Indeterminate More than 3/fspl+1/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal
INT5	$\overline{INT5}$	IMF AND EF8 = 1	Falling edge	Less than 1/fcgck: Noise More than 1/fcgck and less than 2/fcgck: Indeterminate More than 2/fcgck: Signal	Less than 4/fs: Noise More than 4/fs and less than 8/fs: Indeterminate More than 8/fs: Signal

Note 1: fcgck, Gear clock [Hz]; fs, low frequency clock [Hz]; fspl, Sampling interval [Hz]

4.3.1 Low power consumption function

External interrupts have a function that saves power by using the low power consumption register (POFFCR3) when they are not used.

Setting POFFCR3<INTxEN> to "0" stops (disables) the basic clock for external interrupts and helps save power. Note that this makes external interrupts unavailable. Setting POFFCR3<INTxEN> to "1" supplies (enables) the basic clock for external interrupts and makes external interrupts available.

After reset, POFFCR3<INTxEN> is initialized to "0" and external interrupts become unavailable. When using the external interrupt function for the first time, be sure to set POFFCR3<INTxEN> to "1" in the initial setting of software (before operating the external interrupt control registers).

Note: Interrupt request signals may be generated when INTxEN is changed. Before changing INTxEN, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

4.3.2 External interrupt 0

External interrupt 0 detects the falling edge of the $\overline{INT0}$ pin and generates interrupt request signals.

In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1/fcgck are removed as noise and pulses of 2/fcgck or more are recognized as signals.

In SLOW/SLEEP mode, pulses of less than 4/fs are removed as noise and pulses of 8/fs or more are recognized as signals.

Downloaded from Arrow.com.

4.3.3 External interrupts 1/2/3

External interrupts 1/2/3 detect the falling edge, the rising edge or both edges of the INT1, INT2 and INT3 pins and generate interrupt request signals.

4.3.3.1 Interrupt request signal generating condition detection function

Select interrupt request signal generating conditions at EINTCRx<INTxES> for external interrupts 1/2/3.

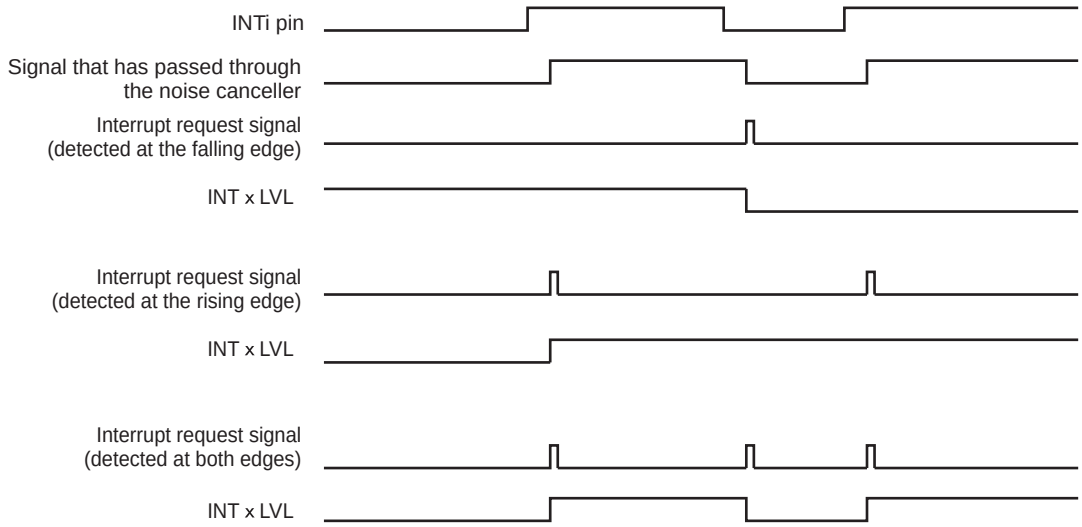
Table 4-2 Selection of Interrupt Request Generation Edge

EINTCRx<INTxES>	Detected at
00	Rising edge
01	Falling edge
10	Both edges
11	Reserved

Note: x=1 to 3

4.3.3.2 A noise canceller pass signal monitoring function when interrupt request signals are generated

The level of a signal that has passed through the noise canceller when an interrupt request is generated can be read by using EINTCRx<INTxLVL>. When both edges are selected as detection edges, the edge where an interrupt is generated can be detected by reading EINTCRx<INTxLVL>.



Note: The contents of EINTCRx<INTxLVL> are updated each time an interrupt request signal is generated.

Figure 4-3 Interrupt Request Generation and EINTCRx<INTxLVL>

4.3.3.3 Noise cancel time selection function

In NORMAL1/2 or IDLE1/2 mode, a signal that has been sampled by fcgck is sampled at the sampling interval selected at EINTCRx<INTxNC>. If the same level is detected three consecutive times, the signal is recognized as a signal. If not, the signal is removed as noise.

Downloaded from Arrow.com.

Table 4-3 Noise Canceller Sampling Lock

EINTCRx<INTxNC>	Sampling interval
00	fcgck
01	fcgck/2^2
10	fcgck/2^3
11	fcgck/2^4

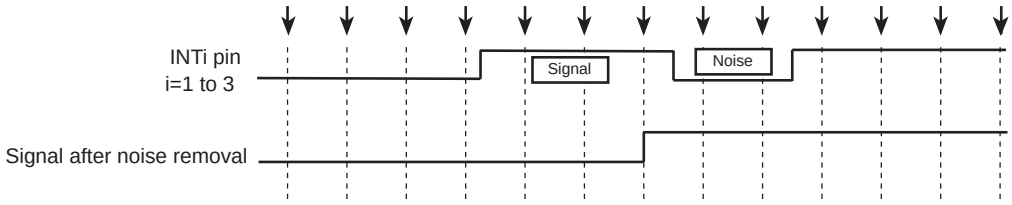


Figure 4-4 Noise Cancel Operation

In SLOW1/2 or SLEEP1 mode, a signal is sampled by the low frequency clock divided by 4. If the same level is detected twice consecutively, the signal is recognized as a signal.

In IDLE0, SLEEP0 or STOP mode, the noise canceller sampling operation is stopped and an external interrupts are unavailable. When operation returns to NORMAL1/2, IDLE1/2, SLOW1/2 or SLEEP1 mode, sampling operation restarts.

Note 1: If noise is input consecutively during sampling of external interrupt pins, the noise cancel function does not work properly. Set EINTCRx<INTxNC> according to the cycle of externally input noise.

Note 2: If an external interrupt pin is used as an output port, the input signal to the port is fixed to "L" when the mode is switched to the output mode, and thus an interrupt request occurs. To use the pin as an output port, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt.

Note 3: Interrupt requests may be generated during transition of the operation mode. Before changing the operation mode, clear the corresponding interrupt enable register to "0" to disable the generation of interrupt. When the operation mode is changed from NORMAL1/2 or IDLE1/2 to SLOW1/2 or SLEEP1, wait 12/fs [s] after the operation mode is changed and clear the interrupt latch. And when the operation mode is changed from SLOW1/2 or SLEEP1 to NORMAL1/2 or IDLE1/2, wait 2/fcgck+3/fspl [s] after the operation mode is changed and clear the interrupt latch.

4.3.4 External interrupt 5

External interrupt 5 detects the falling edge of the $\overline{\text{INT5}}$ pin and generates interrupt request signals.

In NORMAL1/2 or IDLE1/2 mode, pulses of less than 1/fcgck are removed as noise and pulses of 2/fcgck or more are recognized as signals.

In SLOW/SLEEP mode, pulses of less than 4/fs are removed as noise and pulses of 8/fs or more are recognized as signals.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

5. Watchdog Timer (WDT)

The watchdog timer is a fail-safe system to detect rapidly the CPU malfunctions such as endless loops due to spurious noises or the deadlock conditions, and return the CPU to a system recovery routine.

The watchdog timer signals used for detecting malfunctions can be programmed as watchdog interrupt request signals or watchdog timer reset signals.

Note: Care must be taken in system designing since the watchdog timer may not fulfill its functions due to disturbing noise and other effects.

5.1 Configuration

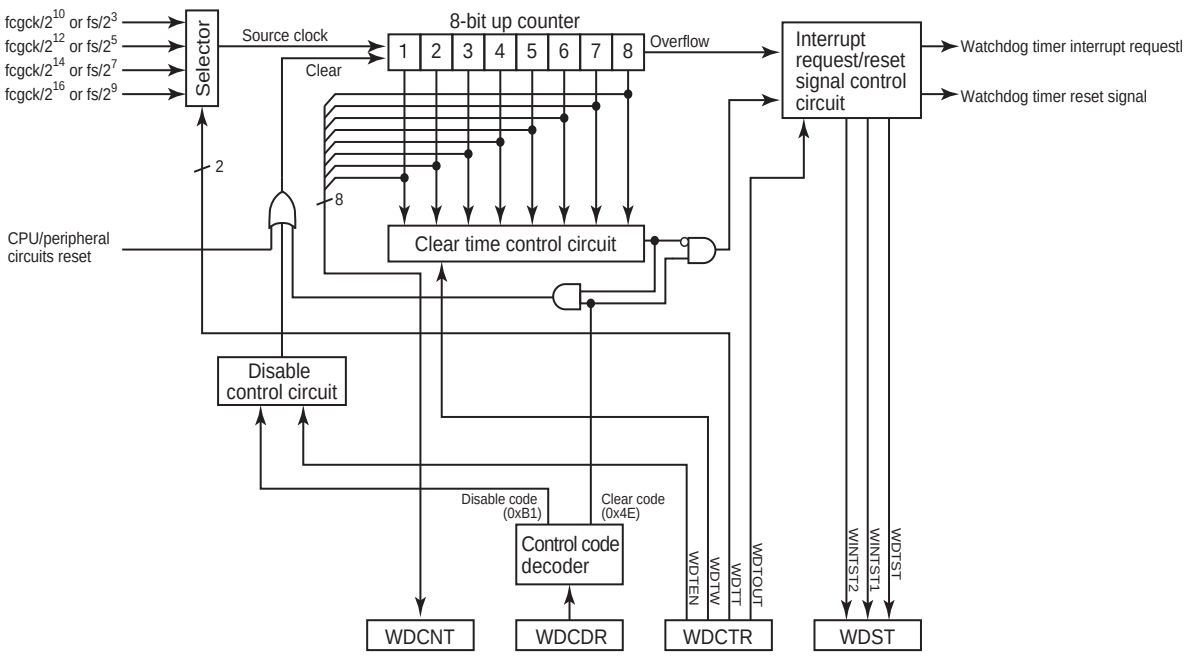


Figure 5-1 Watchdog Timer Configuration

Downloaded from Arrow.com.

5.2 Control

The watchdog timer is controlled by the watchdog timer control register (WDCTR), the watchdog timer control code register (WDCDR), the watchdog timer counter monitor (WDCNT) and the watchdog timer status (WDST).

The watchdog timer is enabled automatically just after the warm-up operation that follows reset is finished.

Watchdog timer control register

WDCTR (0x00FD4)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	WDTEN	WDTW		WDTT		WDTOUT
Read/Write	R	R	R/W	R/W		R/W		R/W
After reset	1	0	1	0	0	1	1	0

WDTEN	Enables/disables the watchdog timer operation.	0 : 1 :	Disable Enable
WDTW	Sets the clear time of the 8-bit up counter.	00 :	The 8-bit up counter is cleared by writing the clear code at any point within the overflow time of the 8-bit up counter.
		01 :	A watchdog timer interrupt request is generated by writing the clear code at a point within the first quarter of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first quarter of the overflow time has elapsed.
		10 :	A watchdog timer interrupt request is generated by writing the clear code at a point within the first half of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first half of the overflow time has elapsed.
		11 :	A watchdog timer interrupt request is generated by writing the clear code at a point within the first three quarters of the overflow time of the 8-bit up counter. The 8-bit up counter is cleared by writing the clear code after the first three quarters of the overflow time have elapsed.

			NORMAL mode		SLOW mode
			DV9CK=0	DV9CK=1	
WDTT	Sets the overflow time of the 8-bit up counter.	00 :	2^{18}/fcgck	2^{11}/fs	2^{11}/fs
		01:	2^{20}/fcgck	2^{13}/fs	2^{13}/fs
		10:	2^{22}/fcgck	2^{15}/fs	2^{15}/fs
		11:	2^{24}/fcgck	2^{17}/fs	2^{17}/fs

WDTOUT	Selects an overflow detection signal of the 8-bit up counter.	0 : 1 :	Watchdog timer interrupt request signal Watchdog timer reset request signal

Note 1: fcgck, Gear clock [Hz]; fs, Low frequency clock [Hz]

Note 2: WDCTR<WDTW>, WDCTR<WDTT> and WDCTR<WDTOUT> cannot be changed when WDCTR<WDTEN> is "1". If WDCTR<WDTEN> is "1", clear WDCTR<WDTEN> to "0" and write the disable code (0xB1) into WDCDR to disable the watchdog timer operation. Note that WDCTR<WDTW>, WDCTR<WDTT> and WDCTR<WDTOUT> can be changed at the same time as setting WDCTR<WDTEN> to "1".

Note 3: Bit 7 and bit 6 of WDCTR are read as "1" and "0" respectively.

Watchdog timer control code register

WDCDR (0x00FD5)

	7	6	5	4	3	2	1	0
Bit Symbol	WDTCR2							
Read/Write	W							
After reset	0	0	0	0	0	0	0	0

WDTCR2	Writes watchdog timer control codes.	0x4E :	Clears the watchdog timer. (Clear code)
		0xB1 :	Disables the watchdog timer operation and clears the 8-bit up counter when WDCTR<WDTEN> is "0". (Disable code)
		Others :	Invalid

Note: WDCDR is a write-only register and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Downloaded from Arrow.com.

8-bit up counter monitor

WDCNT (0x00FD6)

	7	6	5	4	3	2	1	0
Bit Symbol	WDCNT							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0

WDCNT	Monitors the count value of the 8-bit up counter	The count value of the 8-bit up counter is read.

Watchdog timer status

WDST (0x00FD7)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	WINTST2	WINTST1	WDTST
Read/Write	R	R	R	R	R	R	R	R
After reset	0	1	0	1	1	0	0	1

WINTST2	Watchdog timer interrupt request signal factor status 2	0 : 1 :	No watchdog timer interrupt request signal has occurred. A watchdog timer interrupt request signal has occurred due to the overflow of the 8-bit up counter.
WINTST1	Watchdog timer interrupt request signal factor status 1	0 : 1 :	No watchdog timer interrupt request signal has occurred. A watchdog timer interrupt request signal has occurred due to releasing of the 8-bit up counter outside the clear time.
WDTST	Watchdog timer operating state status	0 : 1 :	Operation disabled Operation enabled

Note 1: WDST<WINTST2> and WDST<WINTST1> are cleared to "0" by reading WDST.

Note 2: Values after reset are read from bits 7 to 3 of WDST.

Downloaded from Arrow.com.

5.3 Functions

The watchdog timer can detect the CPU malfunctions and deadlock by detecting the overflow of the 8-bit up counter and detecting releasing of the 8-bit up counter outside the clear time.

The watchdog timer stoppage and other abnormalities can be detected by reading the count value of the 8-bit up counter at random times and comparing the value to the last read value.

5.3.1 Setting of enabling/disabling the watchdog timer operation

Setting WDCTR<WDTEN> to "1" enables the watchdog timer operation, and the 8-bit up counter starts counting the source clock.

WDCTR<WDTEN> is initialized to "1" after the warm-up operation that follows reset is released. This means that the watchdog timer is enabled.

To disable the watchdog timer operation, clear WDCTR<WDTEN> to "0" and write 0xB1 into WDCDR. Disabling the watchdog timer operation clears the 8-bit up counter to "0".

Note: If the overflow of the 8-bit up counter occurs at the same time as 0xB1 (disable code) is written into WDCDR with WDCTR<WDTEN> set at "1", the watchdog timer operation is disabled preferentially and the overflow detection is not executed.

To re-enable the watchdog timer operation, set WDCTR<WDTEN> to "1". There is no need to write a control code into WDCDR.

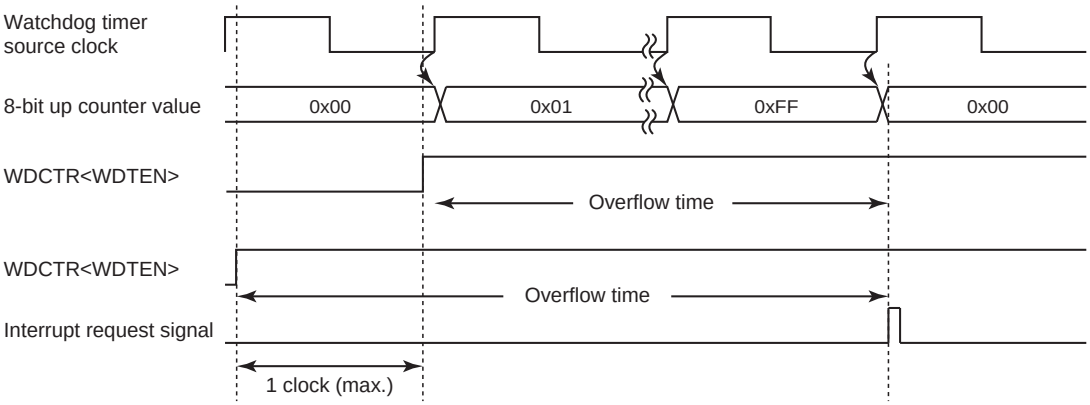


Figure 5-2 WDCTR<WDTEN> Set Timing and Overflow Time

Note: The 8-bit up counter source clock operates out of synchronization with WDCTR<WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR<WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within the period of the overflow time minus 1 source clock cycle.

5.3.2 Setting the clear time of the 8-bit up counter

WDCTR<WDTW> sets the clear time of the 8-bit up counter.

When WDCTR<WDTW> is "00", the clear time is equal to the overflow time of the 8-bit up counter, and the 8-bit up counter can be cleared at any time.

When WDCTR<WDTW> is not "00", the clear time is fixed to only a certain period within the overflow time of the 8-bit up counter. If the operation for releasing the 8-bit up counter is attempted outside the clear time, a watchdog timer interrupt request signal occurs.

At this time, the watchdog timer is not cleared but continues counting. If the 8-bit up counter is not cleared within the clear time, a watchdog timer reset request signal or a watchdog timer interrupt request signal occurs due to the overflow, depending on the WDCTR<WDTOUT> setting.

Downloaded from Arrow.com.

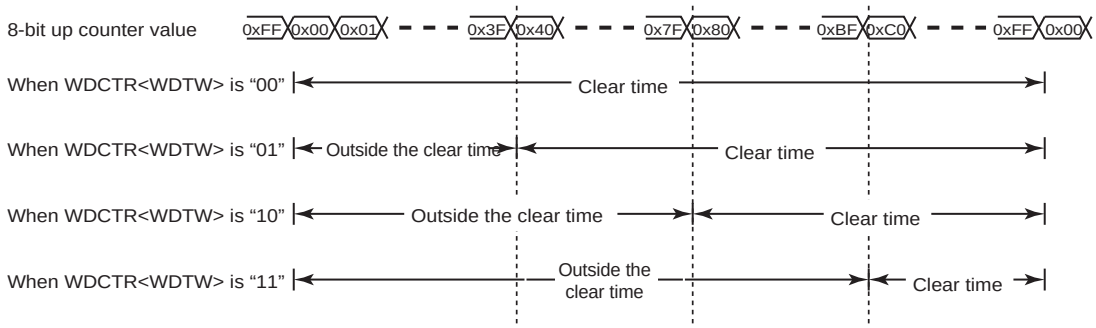


Figure 5-3 WDCTR<WDTW> and the 8-bit up Counter Clear Time

5.3.3 Setting the overflow time of the 8-bit up counter

WDCTR<WDTT> sets the overflow time of the 8-bit up counter.

When the 8-bit up counter overflows, a watchdog timer reset request signal or a watchdog timer interrupt request signal occurs, depending on the WDCTR<WDTOUT> setting.

If the watchdog timer interrupt request signal is selected as the malfunction detection signal, the watchdog counter continues counting, even after the overflow has occurred.

The watchdog timer temporarily stops counting up in the STOP mode (including warm-up) or in the IDLE/SLEEP mode, and restarts counting up after the STOP/IDLE/SLEEP mode is released. To prevent the 8-bit up counter from overflowing immediately after the STOP/IDLE/SLEEP mode is released, it is recommended to clear the 8-bit up counter before the operation mode is changed.

Table 5-1 Watchdog Timer Overflow Time (fcgck=10.0 MHz; fs=32.768 kHz)

WDTT	Watchdog timer overflow time [s]		
	NORMAL mode		SLOW mode
	DV9CK = 0	DV9CK = 1	
00	26.21 m	62.50 m	62.50 m
01	104.86 m	250.00 m	250.00 m
10	419.43 m	1.000	1.000
11	1.678	4.000	4.000

Note: The 8-bit up counter source clock operates out of synchronization with WDCTR<WDTEN>. Therefore, the first overflow time of the 8-bit up counter after WDCTR<WDTEN> is set to "1" may get shorter by a maximum of 1 source clock. The 8-bit up counter must be cleared within a period of the overflow time minus 1 source clock cycle.

5.3.4 Setting an overflow detection signal of the 8-bit up counter

WDCTR<WDTOUT> selects a signal to be generated when the overflow of the 8-bit up counter is detected.

1. When the watchdog timer interrupt request signal is selected (when WDCTR<WDTOUT> is "0")

   Releasing WDCTR<WDTOUT> to "0" causes a watchdog timer interrupt request signal to occur when the 8-bit up counter overflows.

   A watchdog timer interrupt is a non-maskable interrupt, and its request is always accepted, regardless of the interrupt master enable flag (IMF) setting.

Note: When a watchdog timer interrupt is generated while another interrupt, including a watchdog timer interrupt, is already accepted, the new watchdog timer interrupt is processed immediately and the preceding interrupt is put

Downloaded from Arrow.com.

on hold. Therefore, if watchdog timer interrupts are generated continuously without execution of the RETN instruction, too many levels of nesting may cause a malfunction of the microcontroller.

2. When the watchdog timer reset request signal is selected (when WDCTR<WDTOUT> is "1")

   Setting WDCTR<WDTOUT> to "1" causes a watchdog timer reset request signal to occur when the 8-bit up counter overflows.

   This watchdog timer reset request signal resets the TMP89FW24A and starts the warm-up operation.

5.3.5 Writing the watchdog timer control codes

The watchdog timer control codes are written into WDCDR.

By writing 0x4E (clear code) into WDCDR, the 8-bit up counter is cleared to "0" and continues counting the source clock.

When WDCTR<WDTEN> is "0", writing 0xB1 (disable code) into WDCDR disables the watchdog timer operation.

To prevent the 8-bit up counter from overflowing, clear the 8-bit up counter in a period shorter than the overflow time of the 8-bit up counter and within the clear time.

By designing the program so that no overflow will occur, the program malfunctions and deadlock can be detected through interrupts generated by watchdog timer interrupt request signals.

By applying a reset to the microcomputer using watchdog timer reset request signals, the CPU can be restored from malfunctions and deadlock.

Example: When WDCTR<WDTEN> is "0", set the watchdog timer detection time to 2^{20}/fcgck [s], set the counter clear time to half of the overflow time, and allow a watchdog timer reset request signal to occur if a malfunction is detected.

```
                                        LD    (WDCTR), 0y00110011        ;WDTW←10, WDTT←01, WDTOUT←1
Clear the 8-bit up counter at a point after
half of its overflow time and within a peri-
od of the overflow time minus 1 source
clock cycle.
                                        LD    (WDCDR), 0x4E              ;Clear the 8-bit up counter
Clear the 8-bit up counter at a point after
half of its overflow time and within a peri-
od of the overflow time minus 1 source
clock cycle.
                                        LD    (WDCDR), 0x4E              ;Clear the 8-bit up counter
```

Note: If the overflow of the 8-bit up counter and writing of 0x4E (clear code) into WDCDR occur simultaneously, the 8-bit up counter is cleared preferentially and the overflow detection is not executed.

5.3.6 Reading the 8-bit up counter

The counter value of the 8-bit up counter can be read by reading WDCNT.

The stoppage of the 8-bit up counter can be detected by reading WDCNT at random times and comparing the value to the last read value.

5.3.7 Reading the watchdog timer status

The watchdog timer status can be read at WDST.

WDST<WDTST> is set to "1" when the watchdog timer operation is enabled, and it is cleared to "0" when the watchdog timer operation is disabled.

Downloaded from Arrow.com.

WDST<WINTST2> is set to "1" when a watchdog timer interrupt request signal occurs due to the overflow of the 8-bit up counter.

WDST<WINTST1> is set to "1" when a watchdog timer interrupt request signal occurs due to the operation for releasing the 8-bit up counter outside the clear time.

You can know which factor has caused a watchdog timer interrupt request signal by reading WDST<WINTST2> and WDST<WINTST1> in the watchdog timer interrupt service routine.

WDST<WINTST2> and WDST<WINTST1> are cleared to "0" when WDST is read. If WDST is read at the same time as the condition for turning WDST<WINTST2> or WDST<WINTST1> to "1" is satisfied, WDST<WINTST2> or WDST<WINTST1> is set to "1", rather than being cleared.

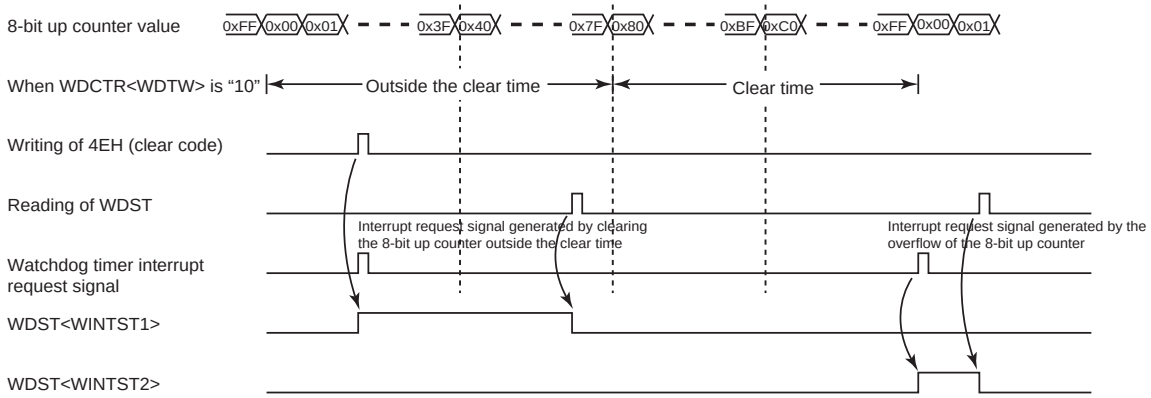


Figure 5-4 Changes in the Watchdog Timer Status

Downloaded from Arrow.com.

Downloaded from Arrow.com.

6. Power-on Reset Circuit

The power-on reset circuit generates a reset when the power is turned on. When the supply voltage is lower than the detection voltage of the power-on reset circuit, a power-on reset signal is generated.

6.1 Configuration

The power-on reset circuit consists of a reference voltage generation circuit and a comparator.

The supply voltage divided by ladder resistor is compared with the voltage generated by the reference voltage generation circuit by the comparator.

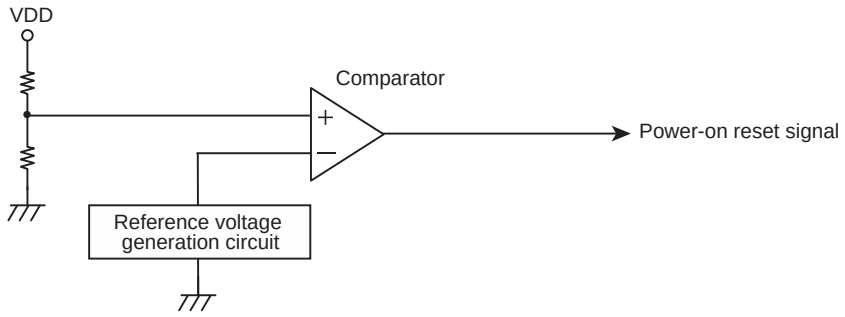


Figure 6-1 Power-on Reset Circuit

6.2 Function

When power supply voltage goes on, if the supply voltage is equal to or lower than the releasing voltage of the power-on reset circuit, a power-on reset signal is generated and if it is higher than the releasing voltage of the power-on reset circuit, a power-on reset signal is released.

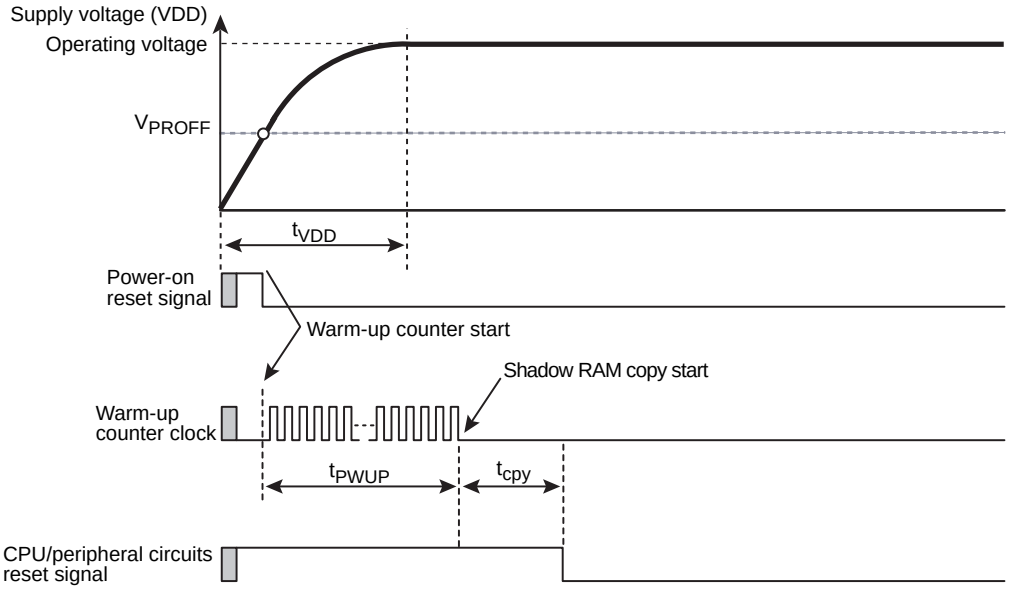
Until the power-on reset signal is generated, a warm-up circuit and a CPU is reset.

When the power-on reset signal is released, the warm-up circuit is activated. The reset of the CPU and peripheral circuits is released after the warm-up time that follows reset release has elapsed.

Increase the supply voltage into the operating range during the period from detection of the power-on reset release voltage until the end of the warm-up time that follows reset release. If the supply voltage has not reached the operating range by the end of the warm-up time that follows reset release, the TMP89FW24A cannot operate properly.

For voltage drop detection, use the voltage detection circuit. For details, refer to the chapter on the voltage detection circuit.

Downloaded from Arrow.com.



Note 1: The power-on reset circuit may operate improperly, depending on fluctuations in the supply voltage (VDD). Refer to the electrical characteristics and take them into consideration when designing equipment.

Note 2: For the AC timing, refer to the electrical characteristics.

Figure 6-2 Operation Timing of Power-on Reset

Downloaded from Arrow.com.

7. Voltage Detection Circuit

The voltage detection circuit detects any decrease in the supply voltage and generates INTVLTD interrupt request signals and voltage detection reset signals.

Note: The voltage detection circuit may operate improperly, depending on fluctuations in the supply voltage (VDD). Refer to the electrical characteristics and take them into consideration when designing equipment.

7.1 Configuration

The voltage detection circuit consists of a reference voltage generation circuit, a detection voltage level selection circuit, a comparator and control registers.

The supply voltage (VDD) is divided by the ladder resistor and input to the detection voltage selection circuit. The detection voltage selection circuit selects a voltage according to the specified detection voltage (VDLVL), and the comparator compares it with the reference voltage. When the comparator detects the selected voltage, a voltage detection reset signal or an INTVLTD interrupt request signal can be generated.

Whether to generate a voltage detection reset signal or an INTVLTD interrupt request signal can be programmed by software. In the former case, a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VDLVL). In the latter case, an INTVLTD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage level.

Note: Since the comparators used for voltage detection do not have a hysteresis structure, INTVLTD interrupt request signals may be generated frequently if the supply voltage (VDD) is close to the detection voltage (VDxLVL). INTVLTD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

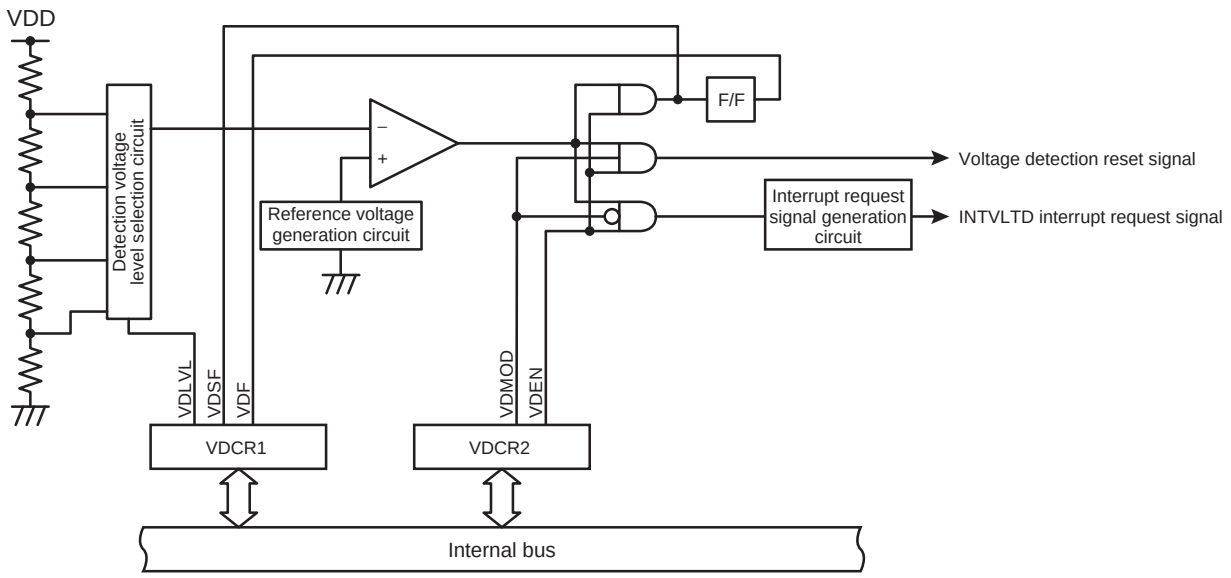


Figure 7-1 Voltage Detection Circuit

Downloaded from Arrow.com.

7.2 Control

The voltage detection circuit is controlled by voltage detection control registers 1 and 2.

Voltage detection control register 1

VDCR1 (0x00FC6)

	7	6	5	4	3	2	1	0
Bit Symbol	"0"		"0"	"0"	VD1F	VD1SF	VD1LVL	
Read/Write	R/W	R	R/W	R/W	R/W	Read Only	R/W	
After reset	0	0	1	0	0	0	0	0

			Read	Write
VDF	Voltage detection flag (Retains the state when VDD < VDLVL is detected)	0:	VDD ≥ VDLVL	Clears VDF to "0"
		1:	VDD < VDLVL	-
VDSF	Voltage detection status flag (Magnitude relation of VDD and VDLVL when they are read)	0:	VDD ≥ VDLVL	
		1:	VDD < VDLVL	
VDLVL	Selection for detection voltage	00:	2.00 +0.15 / -0.15V	
		01:	2.20 +0.15 / -0.15V	
		10:	2.85 +0.15 / -0.15V	
		11:	4.50 +0.20 / -0.20V	

Note 1: VDCR1 is initialized by a power-on reset or an external reset input.

Note 2: When VDF is cleared by the software and is set due to voltage detection at the same time, the setting due to voltage detection is given priority.

Note 3: VDF cannot be programmed to "1" by the software.

Note 4: Bit 7, 5 and 4 of VDCR1 should be cleared to "0".

Voltage detection control register 2

VDCR2 (0x00FC7)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	"0"	"0"	"0"	"0"	VD1MOD	VD1EN
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

VDMOD	Selects the operation mode of voltage detection	0:	Generates an INTVLTD interrupt request signal
		1:	Generates a voltage detection reset signal
VDEN	Enables/disables the operation of voltage detection	0:	Disables the operation of voltage detection
		1:	Enables the operation of voltage detection

Note 1: VDCR2 is initialized by a power-on reset or an external reset input.

Note 2: Bits 7 and 6 of VDCR2 are read as "0".

Note 3: Bit 5 and 2 of VDCR2 should be cleared to "0".

Downloaded from Arrow.com.

7.3 Function

Two detection voltage can be set in the voltage detection circuit. Enabling/disabling the voltage detection and the operation to be executed when the supply voltage (VDD) falls to or below the detection voltage (VDLVL) can be programmed.

7.3.1 Enabling/disabling the voltage detection operation

Setting VDCR2<VDEN> to "1" enables the voltage detection operation. Setting it to "0" disables the operation.

VDCR2<VDEN> is cleared to "0" immediately after a power-on reset or a reset by an external reset input is released.

Note: When the supply voltage (VDD) is lower than the detection voltage (VDLVL), setting VDCR2<VDEN> to "1" generates an INTVLTD interrupt request signal or a voltage detection reset signal at the time.

7.3.2 Selecting the voltage detection operation mode

When VDCR2<VDxMOD> is set to "0", the voltage detection operation mode is set to generate INTVLTD interrupt request signals. When VDCR2<VDxMOD> is set to "1", the operation mode is set to generate voltage detection reset signals.

- When the operation mode is set to generate INTVLTD interrupt signals (VDCR2<VDxMOD>="0")

  When VDCR2<VDxEN>="1", an INTVLTD interrupt request signal is generated when the supply voltage (VDD) falls to the detection voltage (VDxLVL).

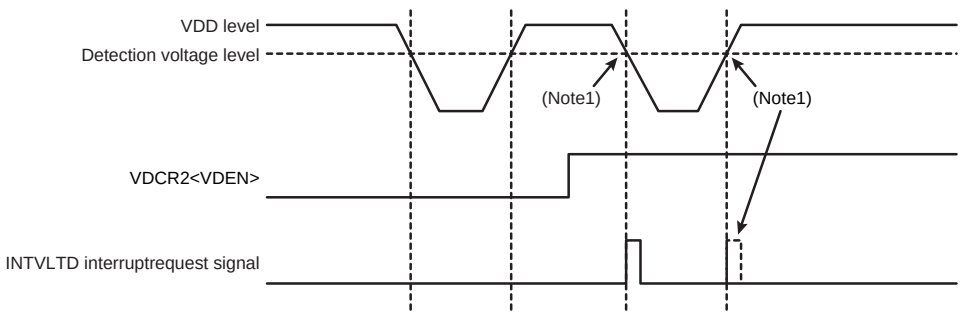


Figure 7-2 Voltage Detection Interrupt Request

Note1: Since the comparators used for voltage detection do not have a hysteresis structure, INTVLTD interrupt request signals may be generated frequently when the supply voltage (VDD) is close to the detection voltage (VDxLVL). INTVLTD interrupt request signals may be generated not only when the supply voltage falls to the detection voltage but also when it rises to the detection voltage.

Note2: If the supply voltage (VDD) falls to the detection voltage (VDxLVL) during IDLE0 or SLEEP0 mode, an INTVLTD interrupt request signal is generated after the TBT counts the specified period and IDLE0 or SLEEP mode is released. In the case of STOP mode, an INTLVTD interrupt request signal is generated after STOP mode is released by the $\overline{\text{STOP}}$ pin.

- When the operation mode is set to generate voltage detection reset signals (VDCR2<VDxMOD>="1")

  When VDCR2<VDxEN> = "1", a voltage detection reset signal is generated when the supply voltage (VDD) becomes lower than the detection voltage (VDxLVL).

  VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. A voltage detection reset signal is generated continuously as long as the supply voltage (VDD) is lower than the detection voltage (VDxLVL).

Downloaded from Arrow.com.

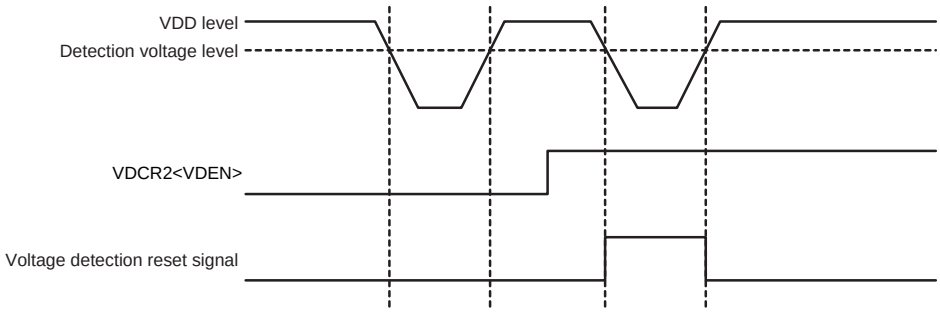


Figure 7-3 Voltage Detection Reset Signal

7.3.3 Selecting the detection voltage level

Select a detection voltage at VDCR1<VDLVL>.

7.3.4 Voltage detection flag and voltage detection status flag

The magnitude relation between the supply voltage (VDD) and the detection voltage (VDLVL) can be checked by reading VDCR1<VDF> and VDCR1<VDSF>.

If VDCR2<VDEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VDLVL), VDCR1<VDF> is set to "1" and is held in this state. VDCR1<VDF> is not cleared to "0" when the supply voltage (VDD) becomes equal to or higher than the detection voltage (VDLVL).

When VDCR2<VDEN> is cleared to "0" after VDCR1<VDF> is set to "1", the previous state is still held. To clear VDCR1<VDF>, "0" must be written to it.

If VDCR2<VDEN> is set at "1", when the supply voltage (VDD) becomes lower than the detection voltage (VDLVL), VDCR1<VDSF> is set to "1". When the supply voltage (VDD) becomes equal to or higher than the detection voltage (VDLVL), VDCR1<VDSF> is cleared to "0".

Unlike VDCR1<VDF>, VDCR1<VDSF> does not hold the set state.

Note 1: When the supply voltage (VDD) becomes lower than the detection voltage (VDLVL) in the STOP, IDLE0 or SLEEP0 mode, the voltage detection flag and the voltage detection status flag are changed after the operation mode is returned to NORMAL or SLOW mode.

Note 2: Depending on the voltage detection timing, the voltage detection status flag (VDSF) may be changed earlier than the voltage detection flag (VDF) by a maximum of 2/fcgck[s].

Downloaded from Arrow.com.

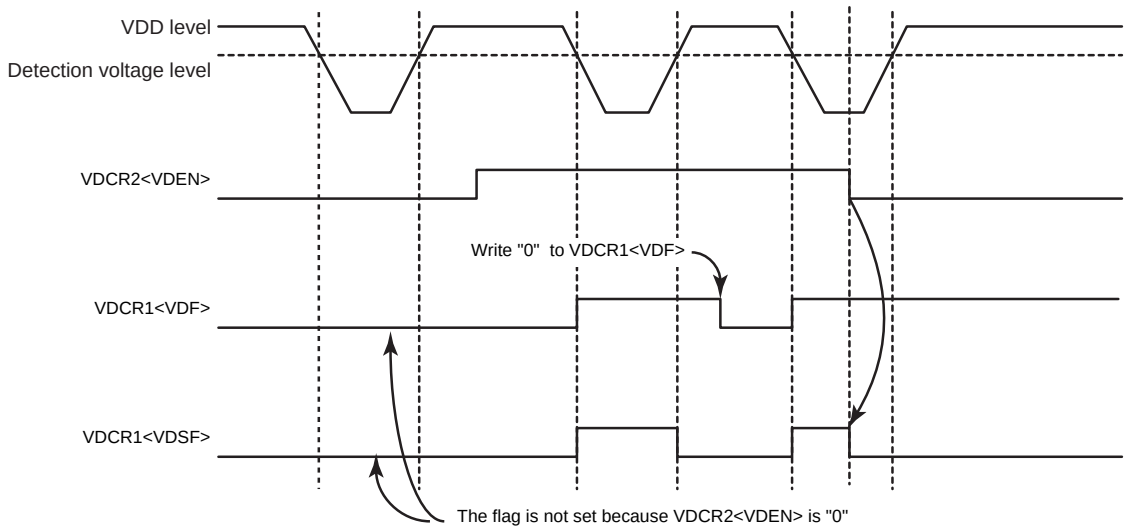


Figure 7-4 Changes in the Voltage Detection Flag and the Voltage Detection Status Flag

Downloaded from Arrow.com.

7.4 Register Settings

7.4.1 Setting procedure when the operation mode is set to generate INTVLTD interrupt request signals

When the operation mode is set to generate INTVLTD interrupt request signal, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Set the detection voltage at VDCR1<VDLVL>.
3. Clear VDCR2<VDMOD> to "0" to set the operation mode to generate INTVLTD interrupt request signals.
4. Set VDCR2<VDEN> to "1" to enable the voltage detection operation.
5. Wait for 5 [μs] or more until the voltage detection circuit becomes stable.
6. Make sure that VDCR1<VDSF> is "0".
7. Clear the voltage detection circuit interrupt latch to "0" and set the interrupt enable flag to "1" to enable interrupts.

Note: When the supply voltage (VDD) is close to the detection voltage (VDxLVL), voltage detection request signals may be generated frequently. If this may pose any problem, execute appropriate wait processing depending on fluctuations in the system power supply and clear the interrupt latch before returning from the INTVLTD interrupt service routine.

To disable the voltage detection circuit while it is enabled with the INTVLTD interrupt request, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Clear VDCR2<VDEN> to "0" to disable the voltage detection operation.

Note: If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

7.4.2 Setting procedure when the operation mode is set to generate voltage detection reset signals

When the operation mode is set to generate voltage detection reset signals, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".
2. Set the detection voltage at VDCR1<VDLVL>.
3. Clear VDCR2<VDMOD> to "0" to set the operation mode to generate INTVLTD interrupt request signals.
4. Set VDCR2<VDEN> to "1" to enable the voltage detection operation.
5. Wait for 5 [μs] or more until the voltage detection circuit becomes stable.
6. Make sure that VDCR1<VDSF> is "0".

Downloaded from Arrow.com.

7. Clear VDCR1<VDF> to "0".

8. Set VDCR2<VDMOD> to "1" to set the operation mode to generate voltage detection reset signals.

Note 1: VDCR1 and VDCR2 are initialized by a power-on reset or an external reset input only. If the supply voltage (VDD) becomes lower than the detection voltage (VDLVL) in the period from release of the voltage detection reset until clearing of VDCR2<VDEN> to "0", a voltage detection reset signal is generated immediately.

Note 2: The voltage detection reset signals are generated continuously as long as the supply voltage (VDD) is lower than the detection voltage (VDLVL).

To disable the voltage detection circuit while it is enabled with the voltage detection reset, make the following setting:

1. Clear the voltage detection circuit interrupt enable flag to "0".

2. Clear VDCR2<VDMOD> to "0" to set the operation mode to generate INTVLTD interrupt request signals.

3. Clear VDCR2<VDEN> to "0" to disable the voltage detection operation.

Note: If the voltage detection circuit is disabled without clearing interrupt enable flag, unexpected interrupt request may occur.

7.5 Caution in the using voltage detevtion circuit

When using the voltage detection circuit, must protect the RESET pin from external noise.

If the RESET pin recognizes as an external reset input, the voltage detection circuit will be initialized and the voltage detection reset signal is released. CPU and the peripheral circuit start operation.

At this time, if supply voltage is below in the level of recommended operational voltage, this may cause malfunction.

Be sure to set a level of RESET pin to high/low by using external circuits.

RESET pin has a built-in pull-up resistor. However when RESET pin is open, it is not effective enough against external noise.

When RESET pin is set to high/low using external circuits, sonfirm a result of the countermesures of blocking external noise.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

8. IO Ports

The TMP89FW24 has 10 parallel input/output ports and 1 input ports(68 pins) as follows:

Table 8-1 List of I/O Ports

Port name	Pin name	Number of pin	Input/output	Seccondary Functions
Port P0	P03 ~ P00	4	Input/output	Also used as the high-frequency oscillator connection pin and the lowfrequency oscillator connection pin
Port P1	P13, P12 ~ P10	3	Input/output	Also used as the external reset input, the Segment output and the external resistor divider output
Port P2	P25 ~ P20	6	Input/output	Also used as the Segment output,external interrupt input,serial interface input/output,the UART input/output, the timer counter input/output, and the onchip debug input.
Port P3	P37 ~ P30	8	Input/output	Also used as the Segment output
Port P4	P47 ~ P40	8	Input/output	Also used as the analog input , the key-on wakeup input,the timercounter input/output,UART input/output,the external interrupt input and the STOP mode release signal input
Port P5	P57 ~ P50	8	Input/output	Also used as the segment output
Port P6	P67 ~ P60	8	Input/output	Also used as the segment output
Port P7	P77 ~ P70	8	Input/output	Also used as the segment output
Port P8	P87 ~ P80	8	Input/output	Also used as the serial bus interface input/output,the UART input/output and the serial interface input/output
Port P9	P97 ~ P90	8	Input/output (P93 input)	Also used as the timer counter input/output, the divider output and the UART input/output

Downloaded from Arrow.com.

Each output port contains a latch, which holds the output data. No input port has a latch, so the external input data should be externally held until the input data is read from outside or reading should be performed several times before processing. Figure 8-1 shows input/output timing examples.

External data is read from an I/O port in the read cycle during execution of the read instruction. This timing cannot be recognized from outside, so that transient input such as chattering must be processed by the program. Data is output to an I/O port in the next cycle of the write cycle during execution of the write instruction.

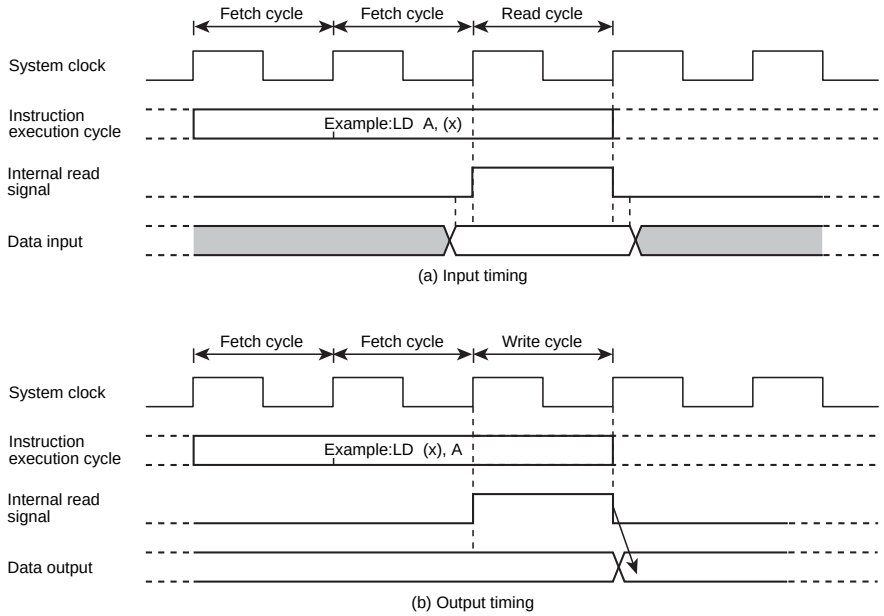


Figure 8-1 Input/Output Timing(Example)

Note: The positions of the read and write cycles may vary, depending on the instruction

2026-04-21
Downloaded from Arrow.com.

8.1 I/O Port Control Registers

The following control registers are used for I/O ports. (The port number is indicated in place of x.) Registers that-can be set vary depending on the port. For details, refer to the description of each port.

- PxDR register

  This is the register for setting output data. When a port is set to the "output mode", the value specified at PxDR is output from the port.

- PxPRD register

  This is the register for reading input data. When a port is set to the "input mode", the current port input status can be read by reading PxPRD.

- PxCR register

  This register switches a port between input and output. A port can be switched between the "input mode" and the "output mode".

- PxFC register

  This register enables the secondary function output of each port. The secondary function output of each port can be enabled or disabled.

- PxOUTCR register

  This register switches the port output between the C-MOS output and the open drain output.

- PxPU register

  This register determines whether or not the built-in pull-up resistor is connected when a port is used in the input mode or as the open drain output.

- PxPUD1 register

  This register determines whether or not the built-in pull-up resistor or pull-down resistor is connected when a port is used in the input mode or as the open drain output.

- PxPUD2 register

  This register determines whether or not the built-in pull-up resistor or the built-in pull-down resistor is connected when PxUD1 resitor set connection.(Only port for embbeded built-in pull-up/built-in pull-down port)

Downloaded from Arrow.com.

8.2 List of I/O Port Settings

For the setting methods for individual I/O ports, refer to the following table.Table 8-2

Table 8-2 List of I/O Port Settings

Port name	Pin name	Function	Register set value				
			PxCR	PxOUTCR	PxFC	PxLCR	Other required settings
Port P0	P03, P01	Port input	0	Without Resister	Without Resister	Without Resister	
		Port output	1				
	P02, P00	Port input	0		0		
		Port output	1		0		
	P03	XTOUT	*		Without Resister		SYSCR2<XTEN>="1"
	P02	XTIN	*		1		SYSCR2<XTEN>="1"
	P01	XOUT	*		Without Resister		SYSCR2<XEN>="1"
	P00	XIN	*		1		SYSCR2<XEN>="1"
Port P1	P13 ~ P12	Port input	0	Without Resister	Without Resister	0	LCDCR2<BRSEL>="1" Note 4
		Port output	1			0	LCDCR2<BRSEL>="1" Note 4
	P10	Port input	0			Without Resister	Note 1
	P10	Port output	1				Note 1
	P13 ~ P12	SEG38 ~ SEG39 output	*			1	LCDCR2<BRSEL>="1" Note 4
	P13 ~ P12	External resitor divider	0			0	LCDCR2<BRSEL>="0" Note 4
	P10	$\overline{\text{RESET}}$ input	*			Without Resister	Note 1
Port P2	P25 ~ P20	Port input	0	*	*	0	
		Port output	1	**	0	0	
	P25 ~ P20	SEG32 ~ SEG37output	*	*	*	1	
	P25	INT3input	0	*	*	0	ITSEL<ITSEL3>="0"
	P24	INTinput	0	*	*	0	ITSEL<ITSEL2>="0"
	P23	INT1input	0	*	*	0	ITSEL<ITSEL1>="0"
	P22	SCLK0 input	0	*	*	0	SERSEL<SRSEL1>="1"
		SCLK0 output	1	**	1	0	SERSEL<SRSEL1>="1"
		TCB0 input	0	*	*	0	SERSEL<SRSEL1>="0" ITSEL<TCBSEL>="1"
		$\overline{\text{PPGB0}}$ output	1	**	1	0	SERSEL<SRSEL1>="0"
	P21	SI0 input	0	*	*	0	SERSEL<SRSEL1>="1"
		RXD0 input	0	*	*	0	SERSEL<SRSEL1>="0"
	P20	SO0 Output	1	**	1	0	SERSEL<SRSEL1>="1"
		TXD0 Output	1	**	1	0	SERSEL<SRSEL1>="0"
Port P3	P37 ~ P30	Portinput	0	Without Resister	Without Resister	0	
		Port output	1			0	
	P37 ~ P30	SEG24 ~ SEG31output	0			1	

Downloaded from Arrow.com.

Table 8-2 List of I/O Port Settings

Port name	Pin name	Function	Register set value				
			PxCR	PxOUTCR	PxFC	PxLCR	Other required settings
Port P4	P47 ~ P40	Port input	0	Without Resister	0	Without Resister	
		Port Output	1		0		
		AIN7 ~ AIN0	0		1		
	P47	$\overline{\text{STOP}}$ input	0		0		
		$\overline{\text{INT5}}$ input	0		0		
	P46	KWI2 input	*		*		KWUCR0<KW2EN>="1"
	P45	RXD2 injput	0		0		SERSEL<SRSEL2>="0"
		KWI1 input	*		*		KWUCR0<KW1EN>="1"
	P44	TXD2 output	1		1		
		KWI0 input	*		*		KWUCR0<KW0EN>="1"
	P43	$\overline{\text{INT0}}$ input	0		0		
	P42	$\overline{\text{PPGB0}}$ output	1		1		
	P41	TCB0 input	0		0		ITSEL<TCBSEL>="0"
Port P5	P57 ~ P50	Port input	0	Without Resister	*	0	
		Port ouput	1		0	0	
		SEG16 ~ SEG23	*		*	1	
Port P6	P67 ~ P60	Port input	0	Without Resister	Without Resister	0	
		Port output	1			0	
		SEG8 ~ SEG15	*			1	
Port P7	P77 ~ P70	Port input	0	Without Resister	Without Resister	0	
		Port output	1			0	
		SEG0 ~ SEG7	*			1	
Port P8	P87 ~ P80	Port input	0	*	*	Without Resister	
		Port output	1	**	0		
	P87	SCLK 0 input	0	*	*		SERSEL<SRSEL1>="0"
		SCLK 0 output	1	**	1		SERSEL<SRSEL1>="0"
	P86	SI0 input	0	*	*		SERSEL<SRSEL1>="0"
		RXD0 input	0	*	*		SERSEL<SRSEL1>="1"
	P85	SO0 output	1	**	1		SERSEL<SRSEL1>="0"
		TXD0 output	1	**	1		SERSEL<SRSEL1>="1"
	P84	RXD1 input	0	*	*		SERSEL<SRSEL3>="0"
	P83	TXD1 output	1	**	1		
	P82	SI1 input	0	*	*		SERSEL<SRSEL0>="0"
	P81	SO1 output	1	**	1		SERSEL<SRSEL0>="0"
		SCL0 input/output	0	1	1		SERSEL<SRSEL0>="1"
	P80	SCLK1input	0	*	*		SERSEL<SRSEL0>="0"
		SCLK1output	1	**	1		SERSEL<SRSEL0>="0"
		SDA0 input/output	0	1	1		SERSEL<SRSEL0>="1"

Downloaded from Arrow.com.

Table 8-2 List of I/O Port Settings

Port name	Pin name	Function	Register set value				
			PxCR	PxOUTCR	PxFC	PxLCR	Other required settings
Port P9	P97 ~ P90	Port input	0	Without Resister	*	Without Resister	
	P97 ~ P94 P92 ~ P90	Port output	1		0		
	P97	TC03 input	0		*		
		INT3 input	0		*		ITSEL<ITSEL3>="1"
		$\overline{PPG03}$ / $\overline{PWM03}$ output	1		1		
	P96	TC02 input	0		*		
		$\overline{PPG02}$ / $\overline{PWM02}$ output	1		1		ITSEL<DVOSEL>="0"
		$\overline{DVO}$ output	1		1		ITSEL<DVOSEL>="1"
	P95	TC01 input	0		*		
		INT2 input	0		*		ITSEL<ITSEL2>="1"
		$\overline{PPG01}$ / $\overline{PWM01}$	1		1		
	P94	TC00 input	0		*		
		INT1 input	0		*		ITSEL<ITSEL1>="1"
		$\overline{PPG00}$ / $\overline{PWM00}$ output	1		1		
	P93	$\overline{EMG}$ input	0		*		SERSEL<SRSEL3>="0"
		RXD1 input	0		*		SERSEL<SRSEL3>="1"
	P92	$\overline{PPGC02}$ output	1		1		SERSEL<SRSEL3>="0"
		TXD1 output	1		1		SERSEL<SRSEL3>="1"
	P91	$\overline{PPGC01}$ output	1		1		SERSEL<SRSEL2>="0" ITSEL<TCSEL>="0"
		$\overline{PPGA0}$ output	1		1		SERSEL<SRSEL2>="0" ITSEL<TCSEL>="1"
		TXD2 output	1		1		SERSEL<SRSEL2>="1"
	P90	TCC0input	0		*		SERSEL<SRSEL2>="0"
		TCA0 input	0		*		SERSEL<SRSEL2>="0" SERSEL<TCA0SEL>="00"
		RXD2 input	0		*		SERSEL<SRSEL2>="1"

Note 1: After the power is turend on.pin P10 serves as an external reset input.To use pin P10 as a port,refer to "How to use the external reset pin as a port"

Note 2: About SERSEL,please refer to "X.X Peripheral input/output select function

Note 3: The symbol and numeric characters in the table have the folowing meanings

Note 4: LCDCR2 should be set when POFFCR2<LCDEN> is "1"

Symbol and numeric characters	Meaning
0	Set "0"
1	Set "1"
*	Don’t care (Operation is the same whether "1" or "0" is selected)
**	The sink open drain output or the C-MOS output can be selected.
Without Register	Ther is no register that corresponds to the bit.

2026-04-21
Downloaded from Arrow.com.

8.3 I/O Port Registers

8.3.1 Port P0 (P03 ~ P00)

Port P0 is a 4-bit input/output port that can be set to input or output for each bit individually, and it is also used as the high-frequency oscillation connection pin and the low-frequency oscillation connection pin.

Port P0 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when the port is used in the input mode.

Table 8-3 Port P0

	-	-	-	-	P03	P02	P01	P00
Secondary function	-	-	-	-	XTOUT	XTIN	XOUT	XIN

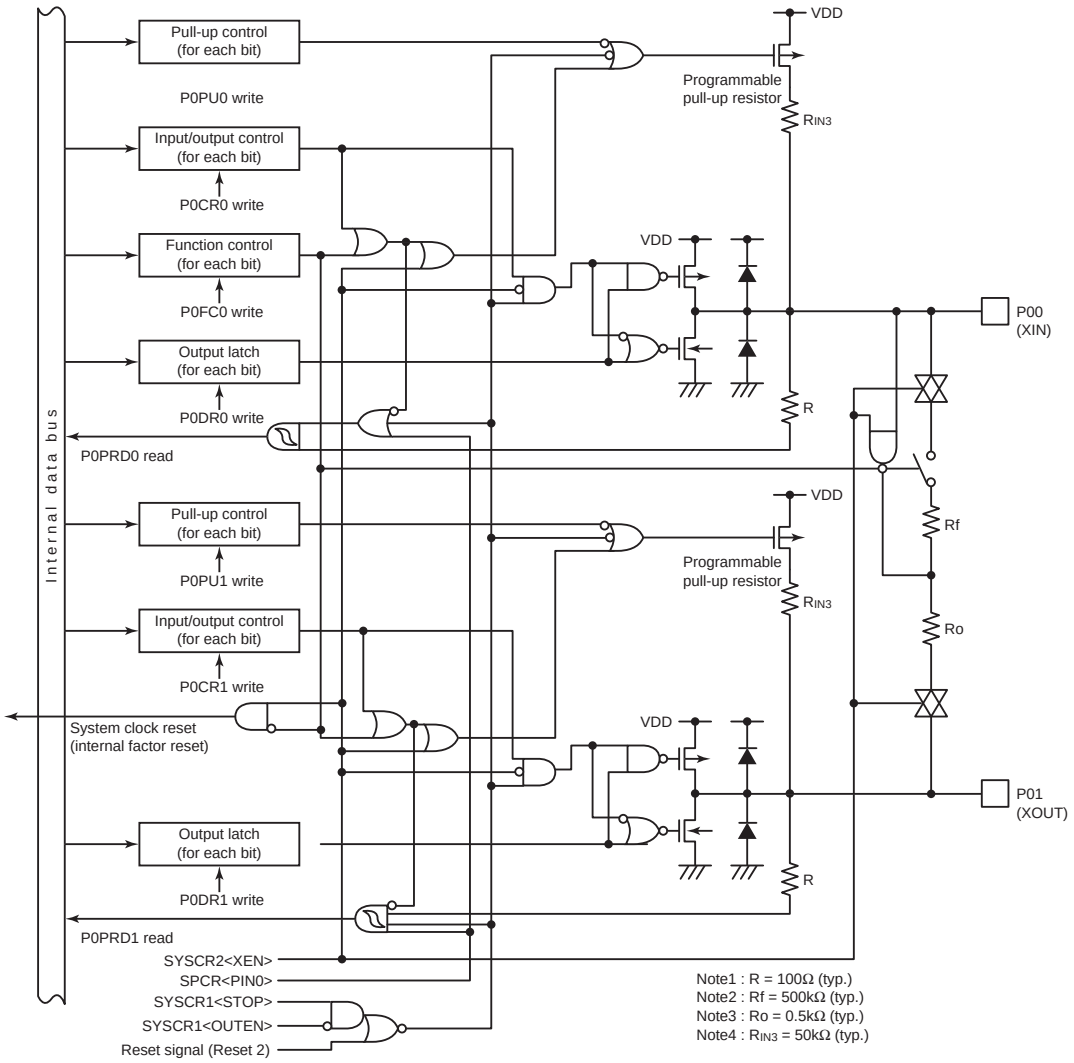


Figure 8-2 Port P0 (P00, P01)

Downloaded from Arrow.com.

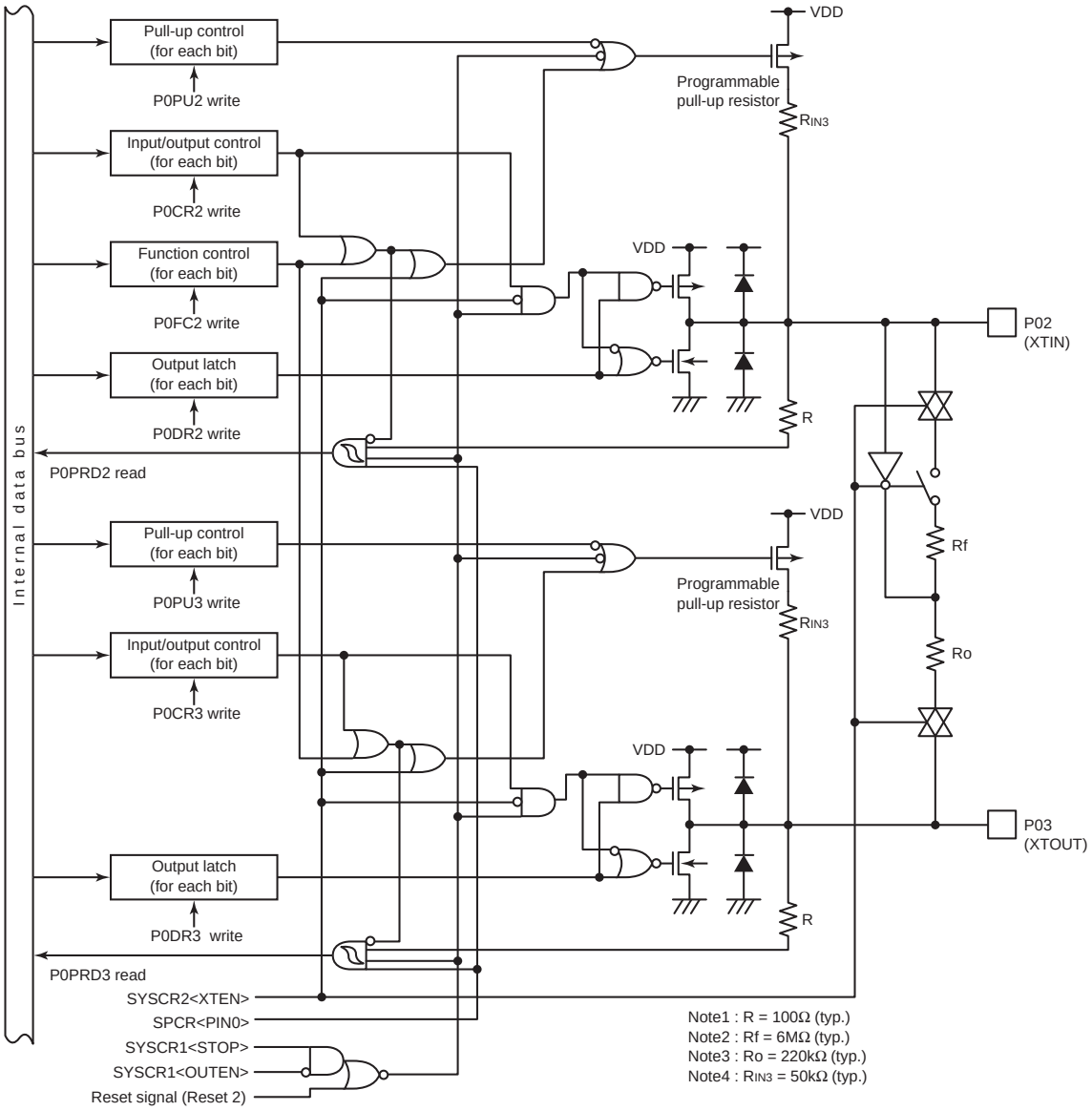


Figure 8-3 Port P0 (P02, P03)

Downloaded from Arrow.com.

Port P0 output latch

P0DR (0x00000)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P03	P02	P01	P00
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Outputs L level when the output mode is selected.			
	1:					Outputs H level when the output mode is selected.			

Port P0 input/output control

P0CR (0x00F1A)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P0CR3	P0CR2	P0CR1	P0CR0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Input mode (port input)			
	1:					Output mode (port output)			

Note: When P0FC0 is set to "1" (XIN(I) is selected as function), P0CR1 and P0CR0 must be clear to "0".If P0CR0 or P0CR1 is set to "1" while P0FC0 is "1" then the output of oscillation port will be short and Operation current will increase.

Port P0 function control

P0FC (0x00F34)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	P0FC2	-	P0FC0
Read/Write		R	R	R	R	R	R/W	R	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:						Port functions		Port functions
	1:						XTIN (I)		XIN (I)

Note 1: To select the external high-freuency clock,the setting"1"to P0FC0 should be done before SYSCR2<XEN> is set to "1"when P0FC0 is "0",the setting"1"to SYSCR2<XEN>generates a system clock(internal factor)reset.

Note 2: Symbol"I"means secondary function input

Port P0 built-in pull-up resistor control

P0PU (0x00F27)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P0PU3	P0PU2	P0PU1	P0PU0
Read/Write		R	R	R	R	R/W	R/W	R/W	R/W
After rset		0	0	0	0	0	0	0	0
Function	0:					The built-in pull-up resistor is not connected.			
	1:					The built-in pull-up resistor is connected. (The resistor is connected in the input mode only. Under any other conditions,setting to "1" does not make the resistor connected.)			

Downloaded from Arrow.com.

Port P0 input data

P0PRD (0x0000D)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	P0PRD3	P0PRD2	P0PRD1	P0PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	*	*	*	*
Function					If the port is in the input mode, the contents of the port are read. If not, "0" is read			

Table 8-4 P0PRD Read Value (P00 ~ P01)

Set condition		P1PRD read value
P0FC0	P0CRi	
*	1	"0"
1	*	"0"
0	0	Contents of port

Note 1: * : Don't care
Note 2: i = 0, 1

Table 8-5 P0PRD Read Value (P02 ~ P03)

Set condition		P0PRD read value
P0FC2	P0CRj	
*	1	"0"
1	*	"0"
0	0	Contents of port

Note 1: * : Don't care
Note 2: j = 2, 3

Downloaded from Arrow.com.

8.3.2 P1 (P13 ~ P10)

Port P1 is a 3-bit input/output port that can be set to input or output for each bit individually, and is also used as the segment output, external resistor divider output and external reset input.Port P1 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when theport is used in the input mode.After reset, pin P10 serves as the external reset input and pin P12,P13 serves as external resitor divider output. To use pin P10 as a port, refer to "How to use external reset input pin as a port.To use pin P12 and P13 as port,set each register of P1 after set POFFCR2<LCDEN> to "1" and set LCDCR2<BRSEL> to "1".

Table 8-6 Port P1

	-	-	-	-	P13	P12	-	P10
Secondary function	-	-	-	-	SEG38 V1	SEG39 V2	-	$\overline{RESET}$

Downloaded from Arrow.com.

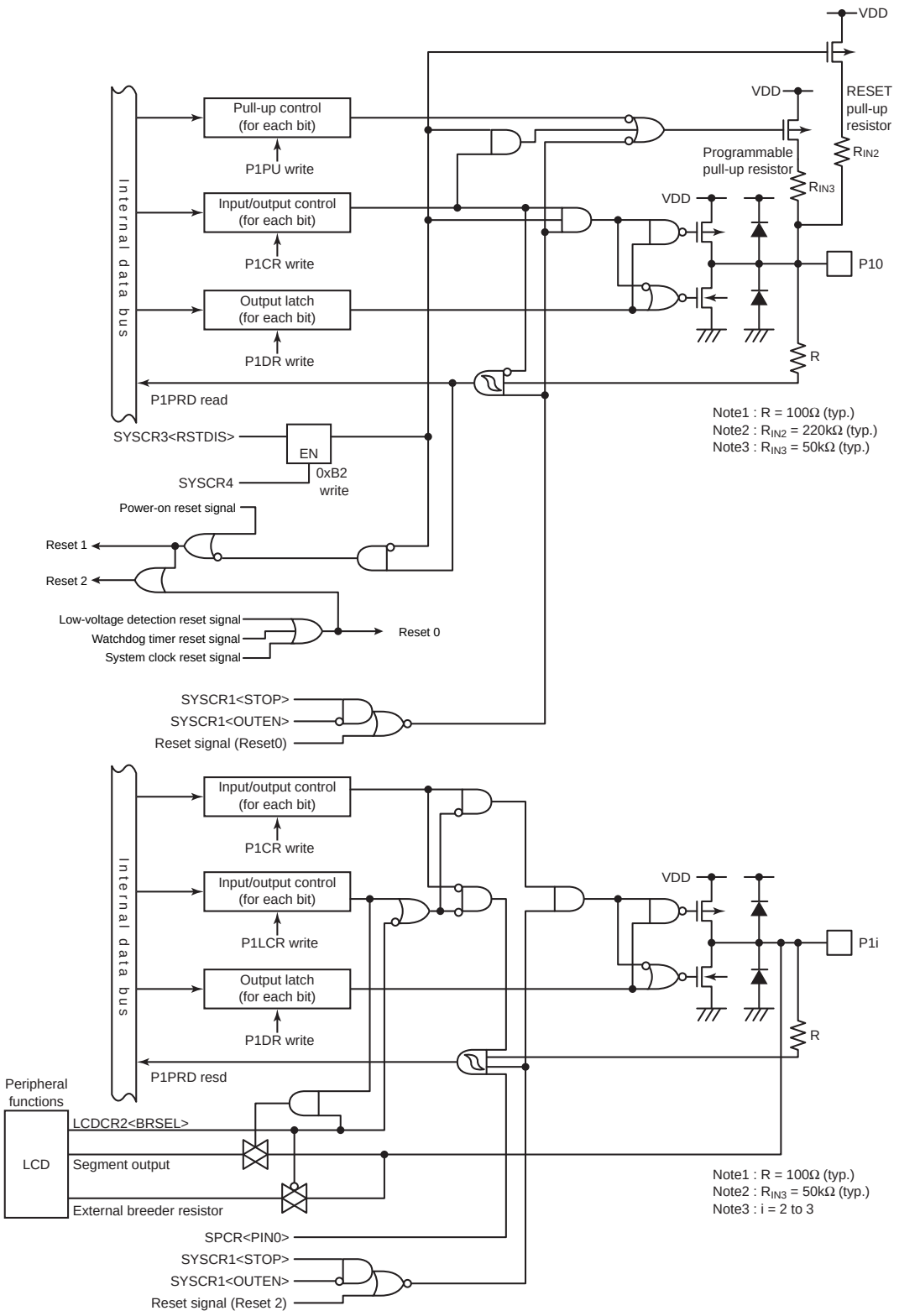


Figure 8-4 Port P1

Downloaded from Arrow.com.

Port P1 output latch

P1DR (0x00001)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P13	P12	-	P10
Read/Write		R	R	R	R	R/W	R/W	R	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Outputs L level when the output mode is selected.			Outputs L level when the output mode is selected.
	1:					Outputs H level when the output mode is selected.			Outputs H level when the output mode is selected.

Port P1 input/output control

P1CR (0x00F1B)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P1CR3	P1CR2	-	P1CR0
Read/Write		R	R	R	R	R/W	R/W	R	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:					Input mode (port input)			Input mode (port input)
	1:					Output mode (port output)			Output mode (port output)

Port P1 built-in pull-up resistor control

P1PU (0x00F28)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	-	-	P1PU0
Read/Write		R	R	R	R	R	R	R	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:								The built-in pull-up resistor is not connected.
	1:								(Note1)

Note 1: The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions,setting to "1" does not make the resistor connected.)

P1 port segment output control

P1LCR (0x00EC1)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	P1LCR3	P1LCR2	-	-
Read/Write		R	R	R	R	R/W	R/W	R	R
After reset		0	0	0	0	0	0	0	0
Function	0:					Input/output port			
	1:					LCD segment output (Note1)			

Note 1: When P1LCR is set to "1",there is no need to set P1CR.

Downloaded from Arrow.com.

Port P1 input data

P1PRD (0x0000E)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	P1PRD3	P1PRD2	-	P1PRD0
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	*	*	0	*
	Function					(Note1)			(Note1)

Note 1: If ther prot is in the input mode,the contents of the port are read.if not ,"0" is read.

Table 8-7 P1PRD Read Value

Set condition		P1PRD read value
P1CRi	P1LCRi	
0	0	Contents of port
1	0	"0"
*	1	"0"

Note 1: * : Don't care
Note 2: i = 0, 2 ~ 3

Downloaded from Arrow.com.

8.3.3 Port P2 (P25 ~ P20)

Port P2 is an 6-bit input/output port that can be set to input or output for each bit individually, and it is also used as LCD segment output, the serial interface input/output, the UART input/output ,the timer/counter input/output and the onchipdebug function.The output circuit has the P-channel output control function and either the sink open drain output or the CMOS output can be selected. Port P2 contains a programmable pull-up/pull-down resistor. This pull-up resistor can be used when the port is used in the input mode or as a sink open drain output.When this port is used as the serial bus interface, the serial interface or the UART, setting for serial interface selecting function is also needed. For details, refer to "8.4 Serial Interface Selecting Function".For the on-chip debug function, refer to the chapter of "On-chip Debug Function (OCD)"

Table 8-8 Port P2

	-	-	P25	P24	P23	P22	P21	P20
Secondary function	-	-	SEG32 INT3	SEG33 INT2	SEG34 INT1	SEG35 SCLK0 TCB0 $\overline{PPGB0}$	SEG36 SI0 RXD0 OCDIO	SEG37 SO0 TXD0 OCDCK

Downloaded from Arrow.com.

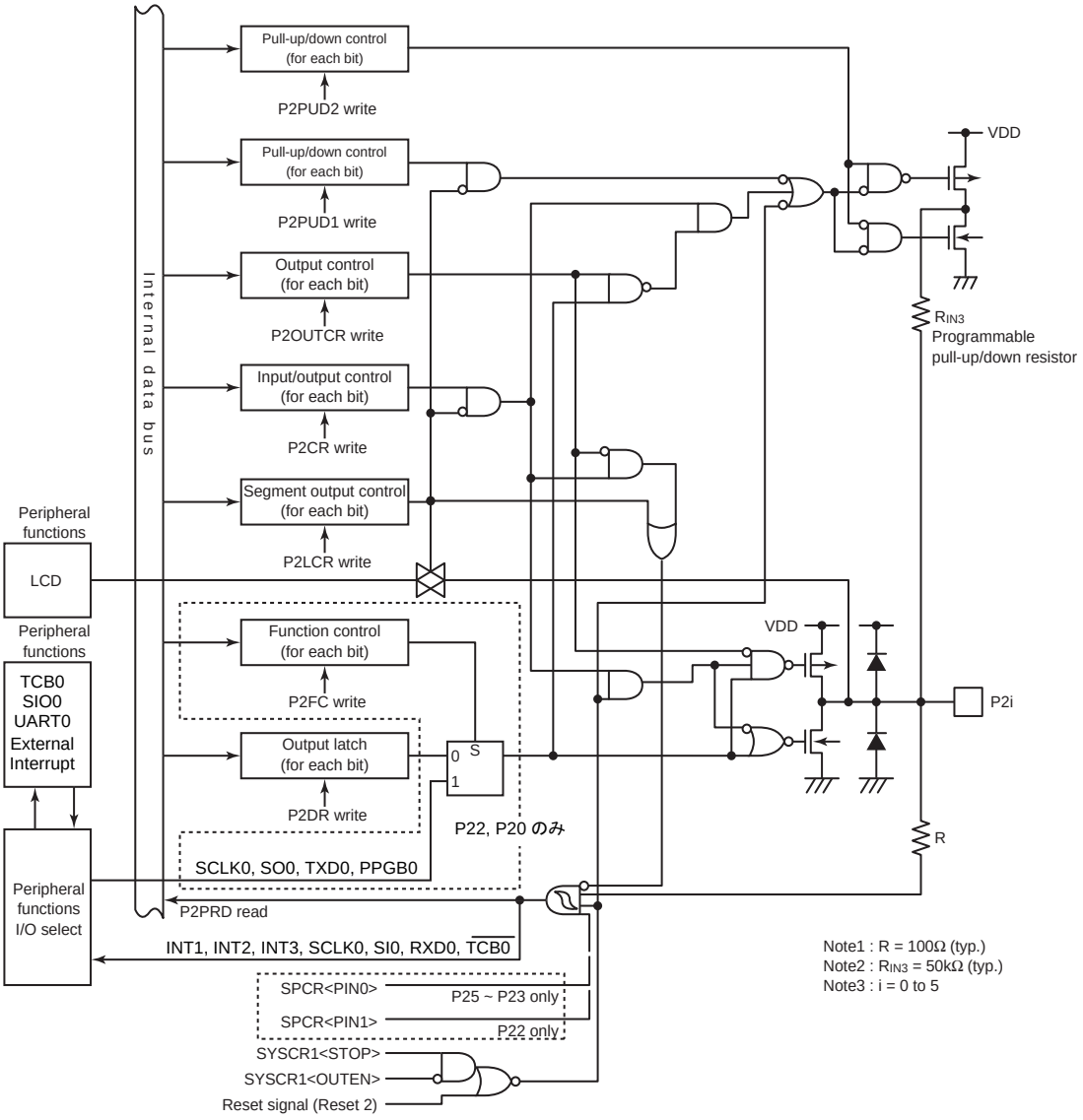


Figure 8-5 Port P2

Downloaded from Arrow.com.

Port P2 output latch

P2DR (0x00002)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	P25	P24	P23	P22	P21	P20
Read/Write		R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:			Outputs L level when the output mode is selected					
	1:			Outputs H level when the output mode is selected. (Serves as Hi-Z or pull-up depending on settings of P2OUTCRand P2PU.)					

Port P2 function control

P2CR (0x00F1C)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	P2CR5	P2CR4	P2CR3	P2CR2	P2CR1	P2CR0
Read/Write		R	R	R/W	R/W	R/W	R/W	R/W	R/W
After rset		0	0	0	0	0	0	0	0
Function				input mode(Port input)					
	0:			INT3 (I)	INT2 (I)	INT1 (I)	SCLK0 (I) TCB0 (I)	RXD0 (I) SI0 (I)	-
				Output mode(Port output)					
	1:			-	-	-	SCLK0 (O) $\overline{\text{PPGB0}}$ (O)	-	TXD0 (O) SO0 (O)

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output. Symbol "I/O" mean-ssecondary function input/output

P2 Function control

P2FC (0x00F36)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	-	-	-	P2FC2	-	P2FC0
Read/Write		R	R	R	R	R	R/W	R	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:						Port function		POrt function
	1:						SCLK0 (O) $\overline{\text{PPGB0}}$ (O)		TXD0 (O) SO0 (O)

Port P2 output control

P2OUTCR (0x00F43)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	P2OUT5	P2OUT4	P2OUT3	P2OUT2	P2OUT1	P2OUT0
Read/Write		R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:			C-MOS output					
	1:			open drain output					

Note: I/O : secondary function input/output

Downloaded from Arrow.com.

Port P2 built-in pull-up/pull-down resistor control

P2PUD1
(0x00F29)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	P2PUD15	P2PUD14	P2PUD13	P2PUD12	P2PUD11	P2PUD10
Read/Write		R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:			The built-in pull-up resistor is not connected.					
	1:			The built-in pull-up resistor is connected.(The resistor is connected only when the port is used in the input mode or as the open drain output. Under any other conditions.setting to "1" does not make the resitor connected)					

Port P2 built-in pull-up/pull-dwon select control

P2PUD2
(0x00ED2)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	P2PUD25	P2PUD24	P2PUD23	P2PUD22	P2PUD21	P2PUD20
Read/Write		R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	1	1	1	1	1	1
Function	0:			The built-in pull-down resistor is selected.					
	1:			The built-in pull-up resistor is selected.					

Note: Select pull-up/pull-down resistor(P2PUD2) after disconnect pull-up/pull-down resitor by built-in pull-up/pull-down resitor control (P2PUD1)

Port P2 output control

P2LCR
(0x00EC2)

		7	6	5	4	3	2	1	0
Bit Symbol		-	-	P2LCR5	P2LCR4	P2LCR3	P2LCR2	P2LCR1	P2LCR0
Read/Write		R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:			input/output port					
	1:			LCD Segment oputput Note 1					

Note 1: When P2LCR is set to "1", the value set to P2FC,P2CR and P2OUTCR has no meaning.

Downloaded from Arrow.com.

Port P2 input data

P2PRD (0x0000F)	7	6	5	4	3	2	1	0
Bit Symbol	R	R	P2PRD5	P2PRD4	P2PRD3	P2PRD2	P2PRD1	P2PRD0
Read/Write	0	0	R	R	R	R	R	R
After reset			*	*	*	*	*	*
Function			If the port is used in the input mode or as the open drain output, the contents of the port are read. If not, "0" is read.					

Table 8-9 P2PRD Read Value (P20 ~ P25)

Set condtion		P2PRD read value
P2CRi	P2OUTCRi	
0	*	Contents of port
1	0	"0"
1	1	Contents of port

Note 1: * : Don't care

Note 2: i = 0 ~ 5

Downloaded from Arrow.com.

8.3.4 Port P3 (P37 ~ P30)

Port P3 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the LCD segment output.Port P3 contains a programmable pull-up/pull-down resistor. This pull-up/pulldown resistor can be used when theport is used in the input mode.

Table 8-10 Port P3

	P37	P36	P35	P34	P33	P32	P31	P30
Secondary function	SEG24	SEG25	SEG26	SEG27	SEG28	SEG29	SEG30	SEG31

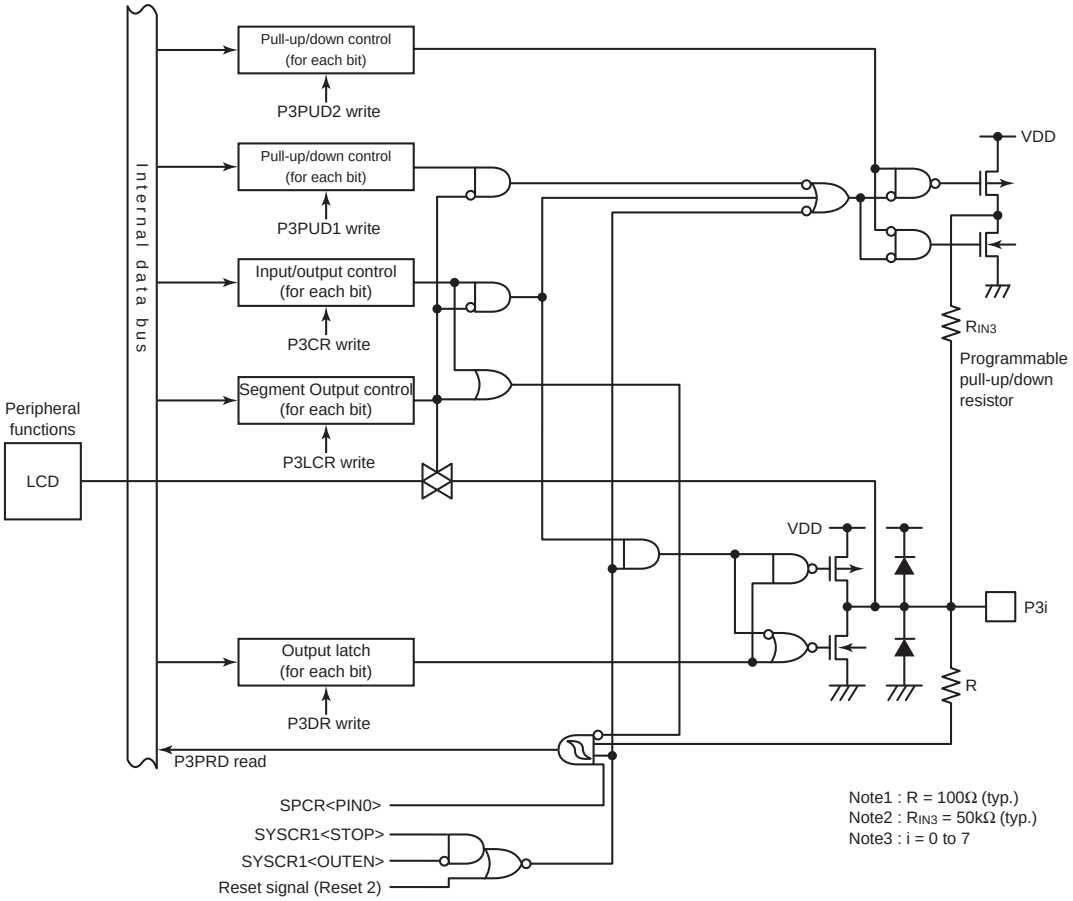


Figure 8-6 Port P3

Downloaded from Arrow.com.

Port P3 output latch

P3DR (0x0003)

		7	6	5	4	3	2	1	0
Bit Symbol		P37	P36	P35	P34	P33	P32	P31	P30
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Afrter reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected.							

P3 input/output control

P3CR (0x0F1D)

		7	6	5	4	3	2	1	0
Bit Symbol		P3CR7	P3CR6	P3CR5	P3CR4	P3CR3	P3CR2	P3CR1	P3CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode (port input)							
	1:	Output mode (port output)							

P3 built-in pull-up/pull-down resistor control

P3PUD1 (0x00F2A)

		7	6	5	4	3	2	1	0
Bit Symbol		P3PUD17	P3PUD16	P3PUD15	P3PUD14	P3PUD13	P3PUD12	P3PUD11	P3PUD10
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	The built-in pull-up/pull-down resistor is not connected.							
	1:	The resistor is connected (only the port is used in the input mode or the port is used in the open drain mode. Under any other conditions, setting to "1" does not make the resistor connected.)							

P3 built-in pull-up/pull-down select control

P3PUD2 (0x00ED3)

		7	6	5	4	3	2	1	0
Bit Symbol		P3PUD27	P3PUD26	P3PUD25	P3PUD24	P3PUD23	P3PUD22	P3PUD21	P3PUD20
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		1	1	1	1	1	1	1	1
Function	0:	The built-in pull-down resistor is selected.							
	1:	The built-in pull-up resistor is selected.							

Note: Select pull-up/pull-down resistor(P3PUD2) after disconnect pull-up/pull-down resitor by built-in pull-up/pull-down resitor control (P3PUD1)

Port P3 output control

P3LCR (0x00EC3)

		7	6	5	4	3	2	1	0
Bit Symbol		P3LCR7	P3LCR6	P3LCR5	P3LCR4	P3LCR3	P3LCR2	P3LCR1	P3LCR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	input/output port							
	1:	LCD Segment output (Note1)							

Note 1: When P3LCR is set to the "1", the value set to P3CR has no meaning.

Downloaded from Arrow.com.

Port P3 input data

P3PRD (0x0010)	7	6	5	4	3	2	1	0
Bit Symbol	P3PRD7	P3PRD6	P3PRD5	P3PRD4	P3PRD3	P3PRD2	P3PRD1	P3PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is in the input mode, the contents of the port are read. If not, "0" is read.							

Table 8-11 P3PRDread value

Set condtino P3CRi	P3PRD read value
0	Contents of port
1	"0"

Note 1: * : Don't care
Note 2: i = 0 ~ 7

Downloaded from Arrow.com.

8.3.5 Port P4 (P47 ~ P40)

Port P4 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is Also used as the analog input , the key-on wakeup input,the timer counter input/output,UART input/output,the external interrupt input and the STOP mode release signal input Port P4 contains a programmable pull-up resistor on the VDD side. This pull-up resistor can be used when theport is used in the input mode.

Table 8-12 Port P4

	P47	P46	P45	P44	P43	P42	P41	P40
Secondary function	AIN7 $\overline{STOP}$ $\overline{INT5}$	AIN6 KWI2	AIN5 KWI1 RXD2	AIN4 KWI0 TXD2	AIN3 $\overline{INT0}$	AIN2 $\overline{PPGB0}$	AIN1 TCB0	AIN0

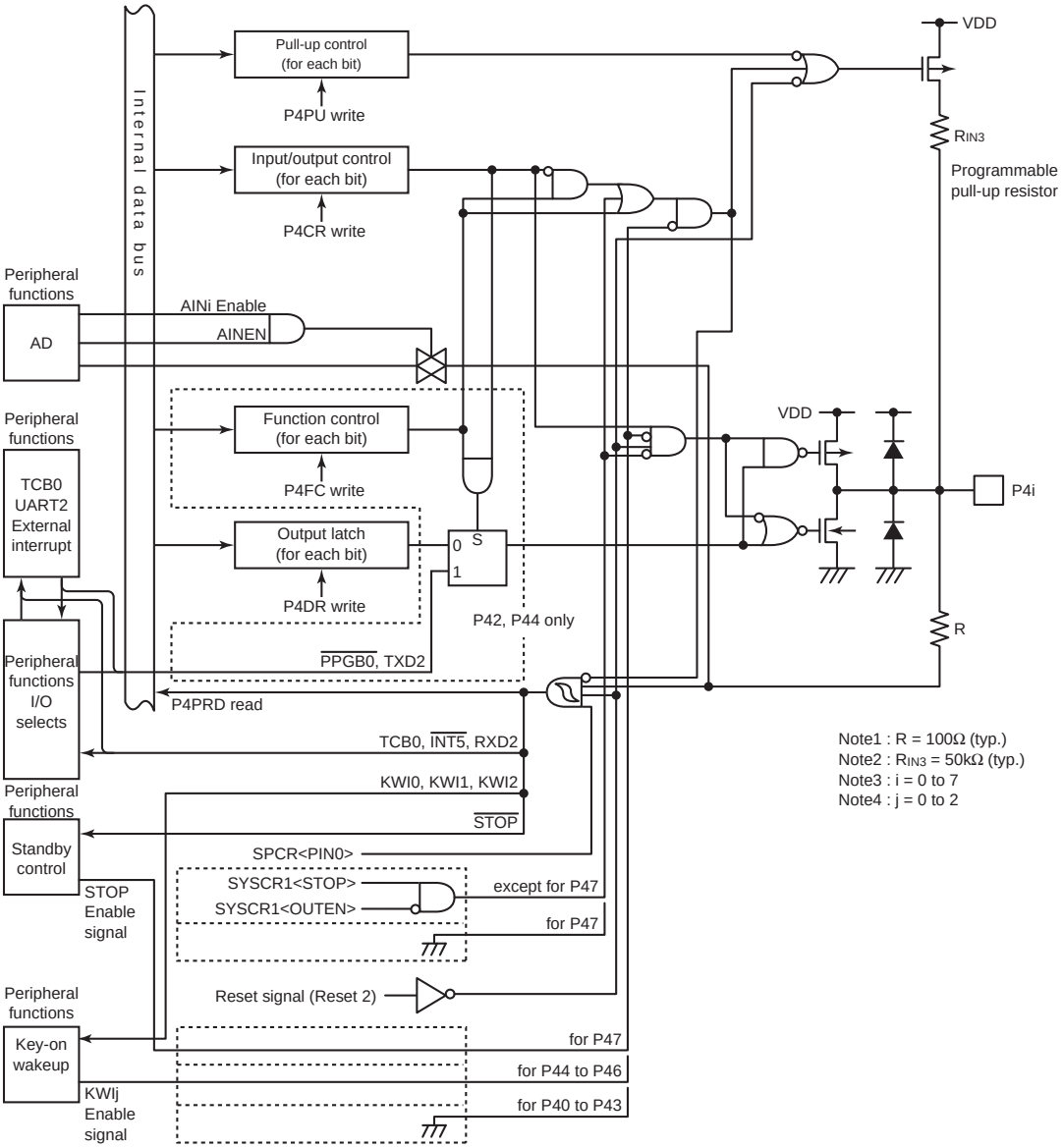


Figure 8-7 Port P4

Downloaded from Arrow.com.

Port P4 output latch

P4DR (0x00004)

		7	6	5	4	3	2	1	0
Bit Symbol		P47	P46	P45	P44	P43	P42	P41	P40
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected.							
	1:	Outputs H level when the output mode is selected.							

Port P4 input/output control

P4CR (0x00F1E)

		7	6	5	4	3	2	1	0
Bit Symbol		P4CR7	P4CR6	P4CR5	P4CR4	P4CR3	P4CR2	P4CR1	P4CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode(Port input)							
		$\overline{\text{STOP}}$ (I) $\overline{\text{INT5}}$ (I)	KWI2 (I)	KWI1 (I) RXD2 (I)	KWI0 (I)	$\overline{\text{INT0}}$ (I)	-	TCB0 (I)	-
	1:	Output mode(Port output)							
		-	-	-	TXD2 (O)	-	$\overline{\text{PPGB0}}$ (O)	-	-

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P4 function control

P4FC (0x00F38)

			7	6	5	4	3	2	1	0
Bit Symbol			P4FC7	P4FC6	P4FC5	P4FC4	P4FC3	P4FC2	P4FC1	P4FC0
Read/Write			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset			0	0	0	0	0	0	0	0
Function	0:		Port Function							
	1:	P4CRx=0	AIN7 (I)	AIN6 (I)	AIN5 (I)	AIN4 (I)	AIN3 (I)	AIN2 (I)	AIN1 (I)	AIN0 (I)
		P4CRx=1	-	-	-	TXD2 (O)	-	$\overline{\text{PPGB0}}$(O)	-	-

P4 built-in pull-up resistor control

P4PU (0x00F2B)

		7	6	5	4	3	2	1	0
Bit Symbol		P4PU7	P4PU6	P4PU5	P4PU4	P4PU3	P4PU2	P4PU1	P4PU0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	The built-in pull-up resistor is not connected.							
	1:	The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in input mode Under any other conditions, setting to "1" does not make the resistor connected.)							

Downloaded from Arrow.com.

Port P4 input data

P4PRD (0x00011)	7	6	5	4	3	2	1	0
Bit Symbol	P4PRD7	P4PRD6	P4PRD5	P4PRD4	P4PRD3	P4PRD2	P4PRD1	P4PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is in the input mode, the contents of the port are read. If not, "0" is read.							

Table 8-13 P4PRD reaed valeu

Set condtion		P4PRD read value
P4CRi	P4FCi	
0	0	contents of port
*	1	"0"
1	*	"0"

Note 1: * : Don't care

Note 2: i = 0 ~ 7

Downloaded from Arrow.com.

8.3.6 Port P5 (P57 ~ P50)

Port P5 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as LCD segment output.

Table 8-14 Port P5

	P57	P56	P55	P54	P53	P52	P51	P50
Secondary function	SEG16	SEG17	SEG18	SEG19	SEG20	SEG21	SEG22	SEG23

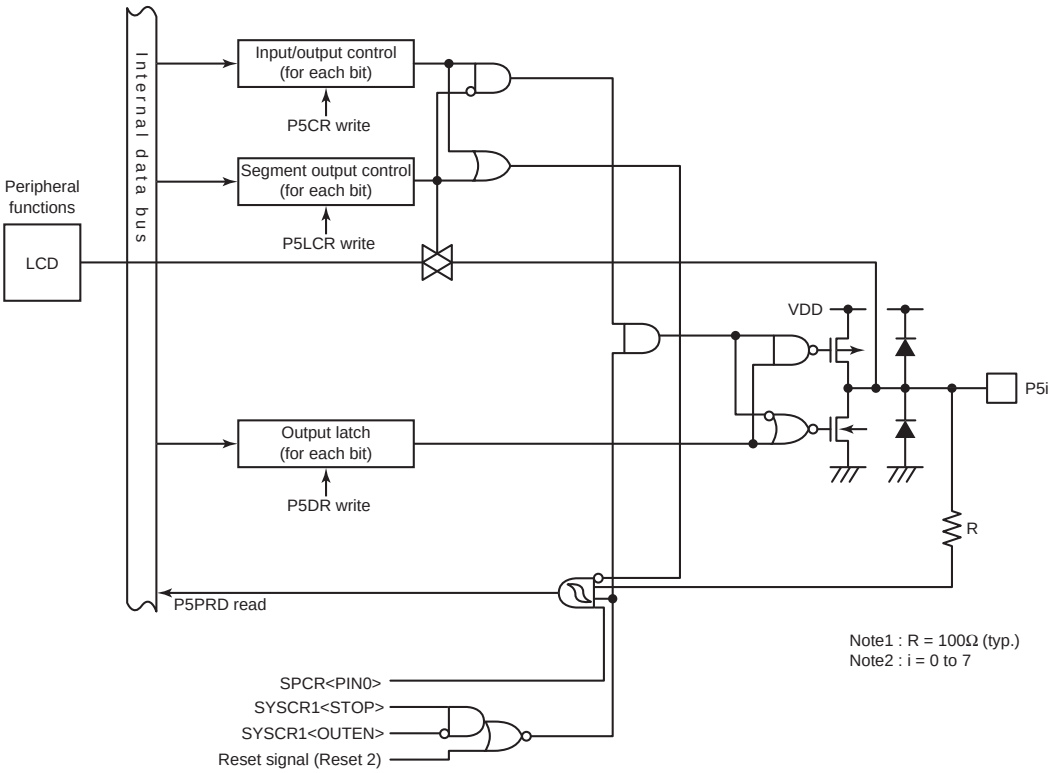


Figure 8-8 Port P5

Downloaded from Arrow.com.

Port P5 output latch

P5DR (0x00005)

		7	6	5	4	3	2	1	0
Bit Symbol		P57	P56	P55	P54	P53	P52	P51	P50
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected							
	1:	Outputs H level when the output mode is selected							

Port P5 input/output control

P5CR (0x00F1F)

		7	6	5	4	3	2	1	0
Bit Symbol		P5CR7	P5CR6	P5CR5	P5CR4	P5CR3	P5CR2	P5CR1	P5CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode(port input)							
	1:	Output mode(port output)							

Port P5 segment output control

P5LCR (0x00EC5)

		7	6	5	4	3	2	1	0
Bit Symbol		P5LCR7	P5LCR6	P5LCR5	P5LCR4	P5LCR3	P5LCR2	P5LCR1	P5LCR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	input/output port							
	1:	LCD Segment output (note1)							

Note 1: When P5LCR is set to "1", the value set to ,P5CR has no meaning.

Port P5 input data

P5PRD (0x00012)

		7	6	5	4	3	2	1	0
Bit Symbol		P5PRD7	P5PRD6	P5PRD5	P5PRD4	P5PRD3	P5PRD2	P5PRD1	P5PRD0
Read/Write		R	R	R	R	R	R	R	R
After reset		*	*	*	*	*	*	*	*
Function		If the port is used in the input mode, the contents of the port are read. If not, "0" is read.							

Table 8-15 P5PRD read value

Set condtion P5CRi	P5PRD read value
0	Contents of port
1	"0"

Note 1: * : Don't care
Note 2: i = 0 ~ 7

Downloaded from Arrow.com.

8.3.7 Port P6 (P67 ~ P60)

Port P6 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as LCD segment output.

Table 8-16 Port P6

	P67	P66	P65	P64	P63	P62	P61	P60
Secondary Function	SEG8	SEG9	SEG10	SEG11	SEG12	SEG13	SEG14	SEG15

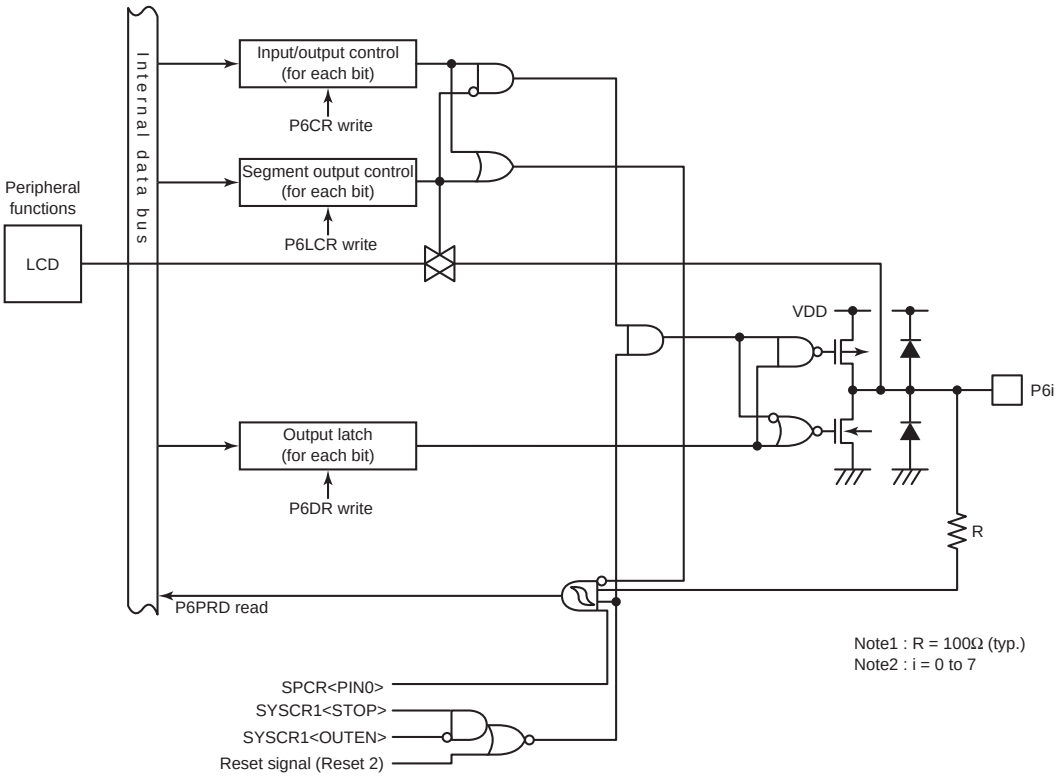


Figure 8-9 Port P6

Downloaded from Arrow.com.

Port P6 output latch

P6DR (0x00006)

		7	6	5	4	3	2	1	0
Bit Symbol		P67	P66	P65	P64	P63	P62	P61	P60
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected							
	1:	Outputs H level when the output mode is selected							

Port 6 input/output control

P6CR (0x00F20)

		7	6	5	4	3	2	1	0
Bit Symbol		P6CR7	P6CR6	P6CR5	P6CR4	P6CR3	P6CR2	P6CR1	P6CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode(Port input)							
	1:	Output mode(Port output)							

Port P6 segment output

P6LCR (0x00EC6)

		7	6	5	4	3	2	1	0
Bit Symbol		P6LCR7	P6LCR6	P6LCR5	P6LCR4	P6LCR3	P6LCR2	P6LCR1	P6LCR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input/output port							
	1:	LCD segment output (Note1)							

Note 1: When P6LCR is set to "1", the value set to ,P6CR has no meaning.

Port P6 input data

P6PRD (0x00013)

	7	6	5	4	3	2	1	0
Bit Symbol	P6PRD7	P6PRD6	P6PRD5	P6PRD4	P6PRD3	P6PRD2	P6PRD1	P6PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is used in the input mode, the contents of the port are read. If not, "0" is read.							

Table 8-17 P6PRD read value

Set condi-tion	P6PRD read value
P6CRi	
0	contents of port
1	"0"

Note 1: * : Don't care

Note 2: i = 0 ~ 7

Downloaded from Arrow.com.

8.3.8 Port P7 (P77 ~ P70)

Port P7 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as LCD segment output.

Table 8-18 port P7

	P77	P76	P75	P74	P73	P72	P71	P70
Secondary function	SEG0	SEG1	SEG2	SEG3	SEG4	SEG5	SEG6	SEG7

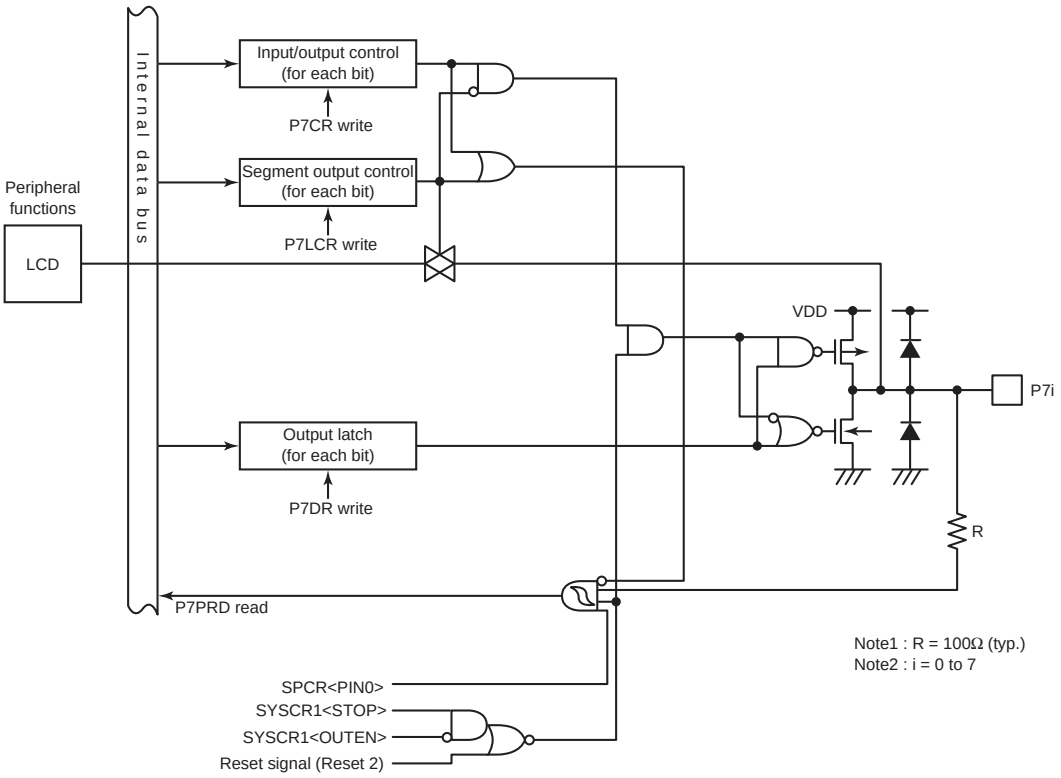


Figure 8-10 Port P7

Downloaded from Arrow.com.

Port P7output latch

P7DR (0x00007)

		7	6	5	4	3	2	1	0
Bit Symbol		P77	P76	P75	P74	P73	P72	P71	P70
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected							
	1:	Outputs H level when the output mode is selected							

Port P7 input/output control

P7CR (0x00F21)

		7	6	5	4	3	2	1	0
Bit Symbol		P7CR7	P7CR6	P7CR5	P7CR4	P7CR3	P7CR2	P7CR1	P7CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode(Port inptut)							
	1:	Output mode(Port output)							

Port P7 segment output control

P7LCR (0x00EC7)

		7	6	5	4	3	2	1	0
Bit Symbol		P7LCR7	P7LCR6	P7LCR5	P7LCR4	P7LCR3	P7LCR2	P7LCR1	P7LCR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	input/output port							
	1:	LCDsegment output (Note1)							

Note 1: When P7LCR is set to "1", the value set to ,P7CR has no meaning.

Port P7 input data

P7PRD (0x00014)

		7	6	5	4	3	2	1	0
Bit Symbol		P7PRD7	P7PRD6	P7PRD5	P7PRD4	P7PRD3	P7PRD2	P7PRD1	P7PRD0
Read/Write		R	R	R	R	R	R	R	R
After reset		*	*	*	*	*	*	*	*
Function		If the port is used in the input mode, the contents of the port are read. If not, "0" is read.							

Table 8-19 P7PRD read value

set con-dtion	P7PRD read value
P7CRi	
0	contents of port
1	"0"

Note 1: * : Don't care
Note 2: i = 0 ~ 7

Downloaded from Arrow.com.

8.3.9 Port P8 (P87 ~ P80)

Port P8 is an 8-bit input/output port that can be set to input or output for each bit individually, and it is also used as the serial bus interface input/output, the serial interface input/output and the UART input/output.The output circuit has the P-channel output control function and either the sink open drain output or the CMOS output can be selected. Port P8 contains a programmable pull-up resistor on the VDD side. This pull-up-resistor can be used when the port is used in the input mode or as a sink open drain output.When this port is used as the serial bus interface, the serial interface or the UART, setting for serial interfaceselecting function is also needed. For details, refer to "X.X Peripheral input/output select function.

Table 8-20 Port P8

	P87	P86	P85	P84	P83	P82	P81	P80
Secondary function	SCLK0	SI0 RXD0	SO0 TXD0	RXD1	TXD1	SI1	SO1 SCL0	SCLK1 SDA0

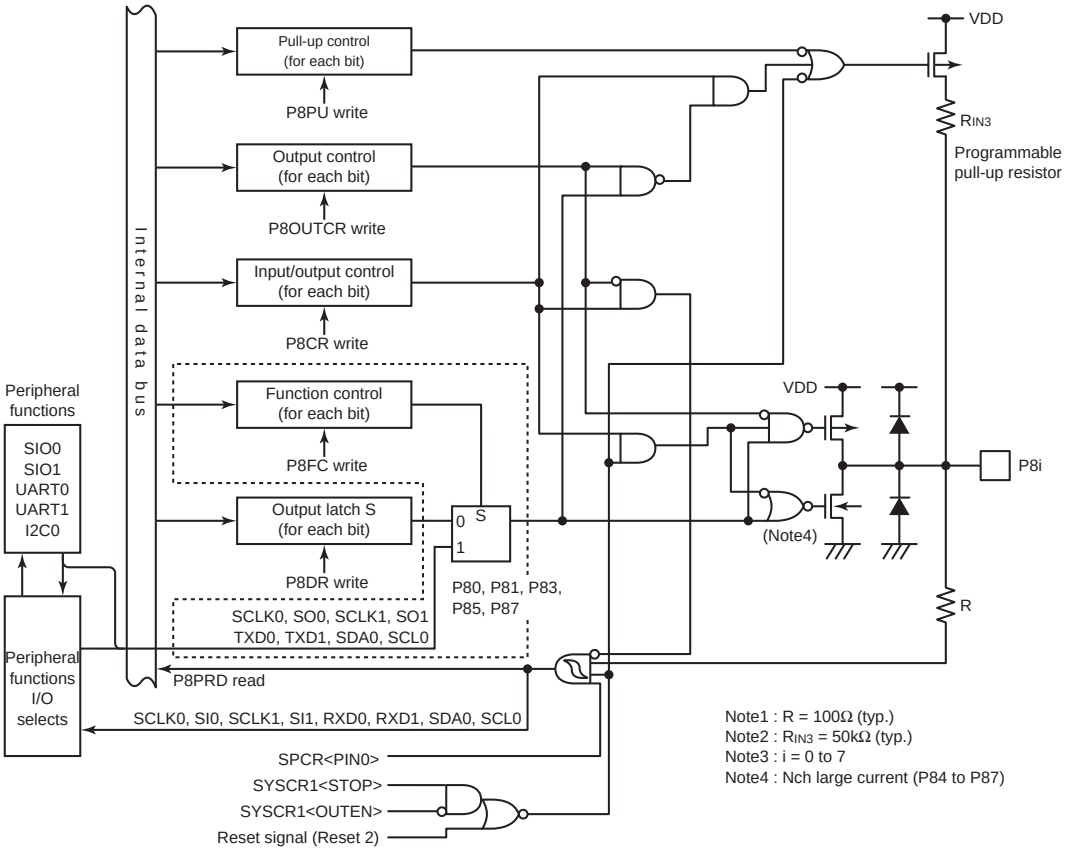


Figure 8-11 Port P8

Downloaded from Arrow.com.

Port P8 output latch

P8DR (0x00008)

		7	6	5	4	3	2	1	0
Bit Symbol		P87	P86	P85	P84	P83	P82	P81	P80
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected							
	1:	Outputs H level when the output mode is selected (Serves as Hi-Z or pull-up depending on settings of P8OUTCR and P8PU)							

Port P8 input/output control

P8CR (0x00F22)

		7	6	5	4	3	2	1	0
Bit Symbol		P8CR7	P8CR6	P8CR5	P8CR4	P8CR3	P8CR2	P8CR1	P8CR0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode(Port input)							
		SCLK0 (I)	SI0 (I) RXD0 (I)	-	RXD1 (I)	-	SI1 (I)	SCL0 (I/O)	SCLK1 (I) SDA0 (I/O)
	1:	OUtpu mode(Port output)							
		SCLK0 (O)	-	SO0 (O) TXD0 (O)	-	TXD1 (O)	-	SO1 (O)	SCLK1 (O)

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P8 function control

P8FC (0x00F3C)

		7	6	5	4	3	2	1	0
Bit Symbol		P8FC7	-	P8FC5	-	P8FC3	-	P8FC1	P8FC0
Read/Write		R/W	R	R/W	R	R/W	R	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Port func-tion		Port func-tion		Port func-tion		Port function	
	1:	SCLK0 (O)		SO0 (O) TXD0 (O)		TXD1 (O)		SO1 (O) SCL0 (I/O)	SCLK1 (O) SDA0 (I/O)

Port P8 output control

P8OUTCR (0x00F49)

		7	6	5	4	3	2	1	0
Bit Symbol		P8OUT5	P8OUT5	P8OUT5	P8OUT4	P8OUT3	P8OUT2	P8OUT1	P8OUT0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Functoin	0:	C-MOS output							
	1:	Open drain output							
		-	-	-	-	-	-	SCL0 (I/O)	SDA0 (I/O)

Downloaded from Arrow.com.

Port P8 built in pull-up control

P8PU (0x00F2F)

		7	6	5	4	3	2	1	0
Bit Symbol		P8PU7	P8PU6	P8PU5	P8PU4	P8PU3	P8PU2	P8PU1	P8PU0
Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	The built-in pull-up resistor is not connected.							
	1:	The built-in pull-up resistor is connected. (The resistor is connected only when the port is used in open drain output or when the port is used in the input mode. Under any other conditions, setting to "1" does not make the resistor connected.)							

Port P8 input data

P8PRD (0x00015)

	7	6	5	4	3	2	1	0
Bit Symbol	P8PRD7	P8PRD6	P8PRD5	P8PRD4	P8PRD3	P8PRD2	P8PRD1	P8PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*
Function	If the port is in the input mode or as the open drain output mode, the contents of the port are read. If not, "0" is read.							

Table 8-21 P8PRD input value

Set condition		P2PRD read value
P8CRi	P8OUTCRi	
0	*	contents of port
1	0	"0"
1	1	contents of port

Note 1: * : Don't care
Note 2: i = 0 ~ 7

Downloaded from Arrow.com.

8.3.10 Port P9 (P97 ~ P90)

Port P9 is an 7-bit input/output port that can be set to input or output for each bit individually and 1-bit input port, and it is also used as the timer counter input/output, the divider output and the UART input/output.

P93 is an input port. When P96 is used as the divider output or P90 and P91 is used as the timer counter output , timer output select functiont is also needed. For details, refer to "1.3 Peripheral input/output select function".

Table 8-22 Port P9

	P97	P96	P95	P94	P93	P92	P91	P90
Secondary function	TC03 $\overline{PPG03}$ $\overline{PWM03}$ INT3	TC02 $\overline{PPG02}$ $\overline{PWM02}$ $\overline{DVO}$	TC01 $\overline{PPG01}$ $\overline{PWM01}$ INT2	TC00 $\overline{PPG00}$ $\overline{PWM00}$ INT1	$\overline{EMG0}$ RXD1	$\overline{PPGC02}$ TXD1	$\overline{PPGC01}$ $\overline{PPGA0}$ TXD2	TCC0 TCA0 RXD2

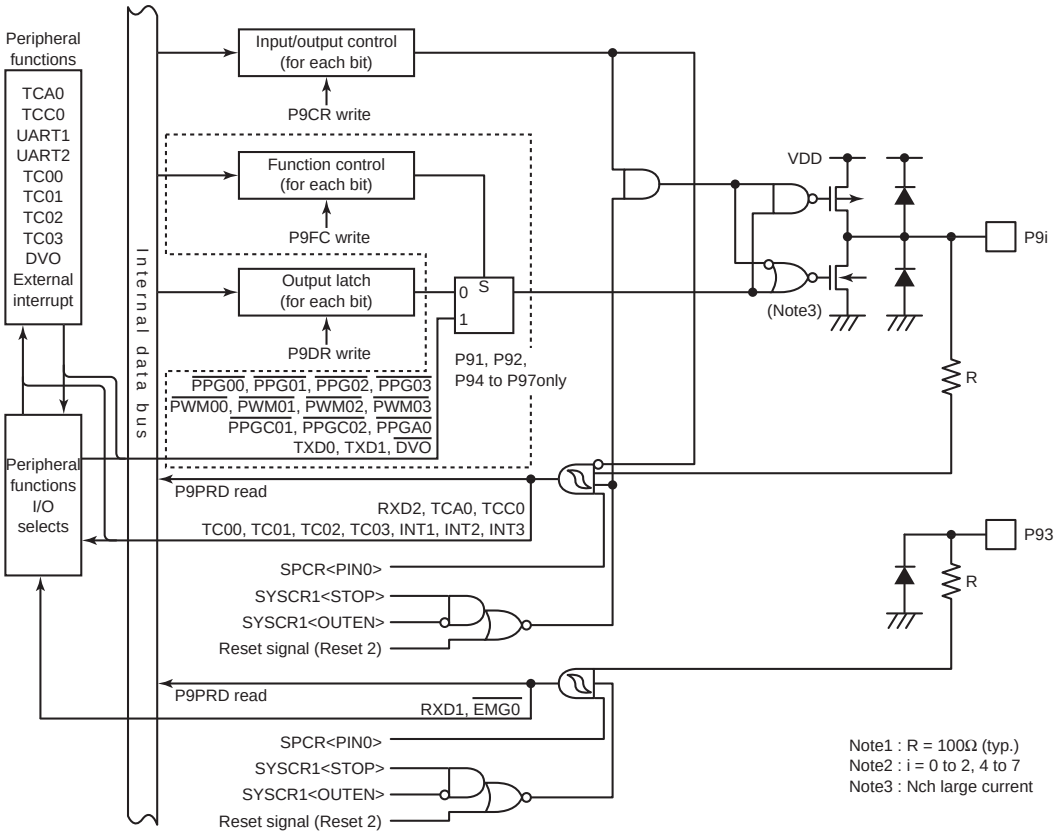


Figure 8-12 Port P9

Downloaded from Arrow.com.

P9 Port output latch

P9DR (0x00009)

		7	6	5	4	3	2	1	0
Bit Symbol		P97	P96	P95	P94	-	P92	P91	P90
Read/Write		R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Outputs L level when the output mode is selected					Outputs L level when the output mode is selected.		
	1:	Outputs H level when the output mode is selected.					Outputs H level when the output mode is selected.		

P9 Port input/output conrol

P9CR (0x00F23)

		7	6	5	4	3	2	1	0
Bit Symbol		P9CR7	P9CR6	P9CR5	P9CR4	-	P9CR2	P9CR1	P9CR0
Read/Write		R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
After reset		0	0	0	0	0	0	0	0
Function	0:	Input mode(Port input)					Input mode(Port input)		
		TC03 (I) INT3 (I)	TC02 (I)	TC01 (I) INT2 (I)	TC00 (I) INT1 (I)		-	-	TCC0 (I) TCA0 (I) RXD2 (I)
	1:	Output mode(Port output)					Output mode(Port output)		
		$\overline{\text{PPG03}}$ (O) $\overline{\text{PWM03}}$ (O)	$\overline{\text{PPG02}}$ (O) $\overline{\text{PWM02}}$ (O) $\overline{\text{DVO}}$ (O)	$\overline{\text{PPG01}}$ (O) $\overline{\text{PWM01}}$ (O)	$\overline{\text{PPG00}}$ (O) $\overline{\text{PWM00}}$ (O)		$\overline{\text{PPGC02}}$ (O) TXD1 (O)	$\overline{\text{PPGC01}}$ (O) $\overline{\text{PPGA0}}$ (O) TXD2 (O)	-

Note: Symbol "I" means secondary function input. Symbol "O" means secondary function output.

Port P9 function control

P9FC (0x00F3D)

		7	6	5	4	3	2	1	0
Bit Symbol		P9FC7	P9FC6	P9FC5	P9FC4	-	P9FC2	P9FC1	-
Read/Write		R/W	R/W	R/W	R/W	R	R/W	R/W	R
After reset		0	0	0	0	0	0	0	0
Function	0:	Port control					Port control		
	1:	$\overline{\text{PPG03}}$ (O) $\overline{\text{PWM03}}$ (O)	$\overline{\text{PPG02}}$ (O) $\overline{\text{PWM02}}$ (O) $\overline{\text{DVO}}$ (O)	$\overline{\text{PPG01}}$ (O) $\overline{\text{PWM01}}$ (O)	$\overline{\text{PPG00}}$ (O) $\overline{\text{PWM00}}$ (O)		$\overline{\text{PPGC02}}$ (O) TXD1 (O)	$\overline{\text{PPGC01}}$ (O) $\overline{\text{PPGA0}}$ (O) TXD2 (O)	

Port P9 input data

P9PRD (0x00016)

	7	6	5	4	3	2	1	0
Bit Symbol	P9PRD7	P9PRD6	P9PRD5	P9PRD4	P9PRD3	P9PRD2	P9PRD1	P9PRD0
Read/Write	R	R	R	R	R	R	R	R
After reset-Function	*	*	*	*	*	*	*	*
Function	If the port is in the input mode, the contents of the port are read. If not, "0" is read.							

Downloaded from Arrow.com.

Table 8-23 P9PRD read value

Set condition P9CRi	P9PRD read value
0	Contents of port
1	"0"

Note 1: * : Don't care
Note 2: i = 0 ~ 7

Downloaded from Arrow.com.

8.4 Peripheral input/output select function

On theTMP89FW24A the Peripheral input/output pins and interrupt source assignment can be changed. Interrupt source assignment can be changed by Peripheral input/output select register1(ITSEL) and peripheral input/output select register2(SERSEL).

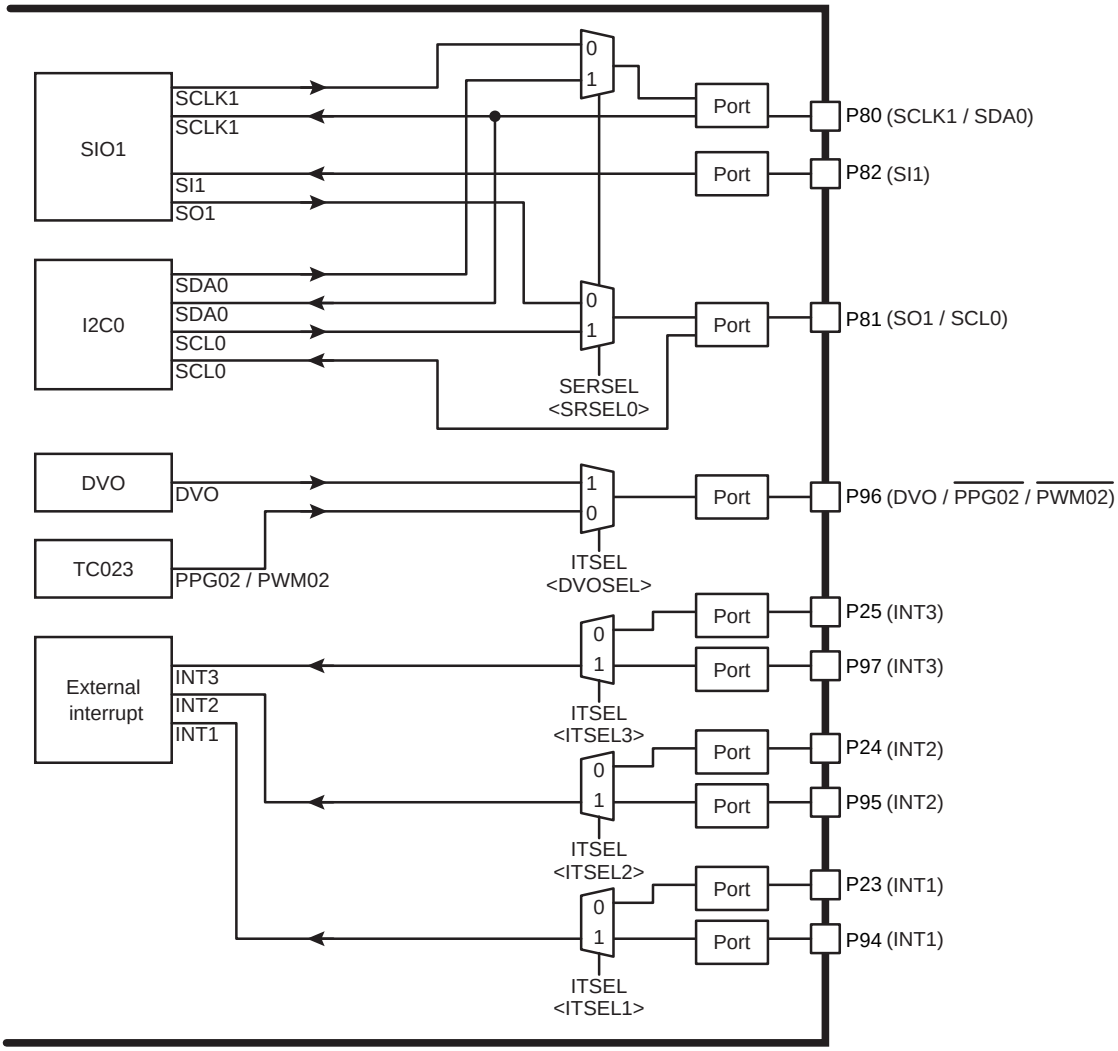


Figure 8-13 Peripheral input/output select

Note: When Peripheral function input/output select register is set ,setting for I/O port is also needed.For detail, refer to the chapter of I/O Ports.

Downloaded from Arrow.com.

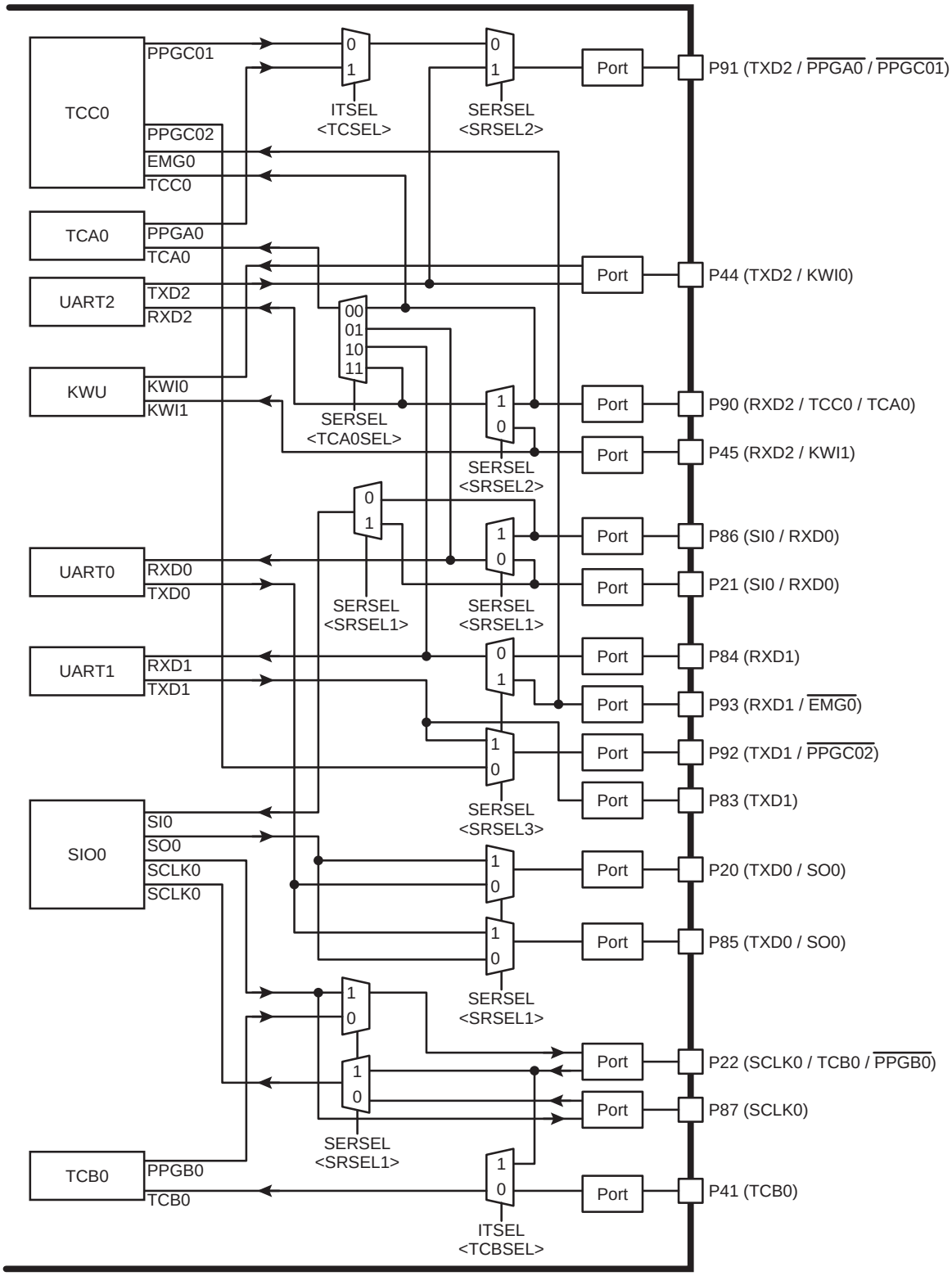


Figure 8-14 Peripheral function input/output select

Note: When Peripheral function input/output select register is set ,setting for I/O port is also needed.For detail, refer to the chapter of I/O Ports.

Downloaded from Arrow.com.

Peripheral function input/output select Register1

ITSEL (0x00FCA)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	TCBSEL	DVOSEL	ITSEL3	ITSEL2	ITSEL1	TCSEL
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TCBSEL	TCB0 input port select	0: 1:	P41input P22 input
DVOSEL	P96 Port ouput select	0: 1:	$\overline{PPG02}$ / $\overline{PWM02}$ output $\overline{DVO}$ output
ITSEL3	INT3 input port assignment	0: 1:	P25 input P97 input
ITSEL2	INT2 input port assignment	0: 1:	P24 input P95 input
ITSEL1	INT1 input port assignment	0: 1:	P23 input P94 input
TCSEL	P91Port output select	0: 1:	$\overline{PPGC01}$ output $\overline{PPGA0}$ output

Note 1: The operation for changing ITSEL must be executed while the applicable peripheral operations are stopped. If ITSEL is switched during operation of these peripheral functions, each peripheral function may receive (transmit) unexpected data and operate improperly.

Note 2: It is recommended to clear the interrupt latch for the applicable peripheral function immediately after changing ITSEL.

Downloaded from Arrow.com.

Peripheral function input/output select Register2

SERSEL (0x00FCB)

	7	6	5	4	3	2	1	0
Bit Symbol	TCA0SEL		-	-	SRSEL3	SRSEL2	SRSEL1	SRSEL0
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TCA0SEL	16-bit timer counter A0 input switching	00:	P90 input (TCA0)					
		01:	RXD0 input (also used as SRSEL1)					
		10:	RXD1 input (also used as SRSEL3)					
		11:	RXD2 input(also used as SRSEL2)					
SRSEL3	Select UART0/TCC0 input/output port		P83	P84	P92	P93		
		0:	TXD1	RXD1	$\overline{PPGC02}$	$\overline{EMG0}$		
		1:	TXD1	-	TXD1	RXD1		
SRSEL2	Select UART2/TCC0/TCA0 input/output port		P90	P91	P44	P45		
		0:	TCC0 / TCA0	$\overline{PPGC01}$ / $\overline{PPGA0}$	TXD2	RXD2		
		1:	RXD2	TXD2	KWI0	KWI1		
SRSEL1	Select UART0/SIO0 input/output port		P20	P21	P22	P85	P86	P87
		0:	TXD0	RXD0	TCB0 / $\overline{PPGB0}$	SO0	SI0	SCLK0
		1:	SO0	SI0	SCLK0	TXD0	RXD0	-
SRSEL0	Select SIO1/SBIO0 input/output port Select IL28(interrupt source)		P80	P81	P82	IL28		
		0:	SCLK1	SO1	SI1	INTSIO1		
		1:	SDA0	SCL0	-	INTSBI0		

Note 1: The operation for changing SERSEL must be executed while the applicable periphera operations are stopped. If SERSEL is switched during operation of these peripheral functions, each peripheral function may receive (transmit) unexpected data and operate improperly.

Note 2: It is recommended to clear the interrupt latch for the applicable peripheral function immediately after changing SERSEL. Interrupt latches are common to INTSIO1 and INTSB10. Therefore, if an interrupt occurs beforeor after SERSEL is switched, it is difficult to tell which function has caused the interrupt

Downloaded from Arrow.com.

Downloaded from Arrow.com.

9. Special Function Registers

The TMP89FW24A adopts the memory mapped I/O system, and all peripheral hardware data control and transfer operations are performed through the special function registers (SFR). SFR1 is mapped on addresses 0x00000 to 0x0003F, SFR2 is mapped on addresses 0x00F00 to 0x00FFF, and SFR3 is mapped on addresses 0x00E40 to 0x00EBF.

9.1 SFR1 (0x00000 to 0x0003F)

Table 9-1 SFR1 (0x00000 to 0x0003F)

Address	Register Name	Address	Register Name
0x00000	P0DR	0x00020	SIO0SR
0x00001	P1DR	0x00021	SIO0BUF
0x00002	P2DR	0x00022	SBI0CR1
0x00003	P3DR	0x00023	SBI0CR2/SBI0SR2
0x00004	P4DR	0x00024	I2C0AR
0x00005	P5DR	0x00025	SBI0DBR
0x00006	P6DR	0x00026	T00REG
0x00007	P7DR	0x00027	T01REG
0x00008	P8DR	0x00028	T00PWM
0x00009	P9DR	0x00029	T01PWM
0x0000A	Reserved	0x0002A	T00MOD
0x0000B	Reserved	0x0002B	T01MOD
0x0000C	Reserved	0x0002C	T001CR
0x0000D	P0PRD	0x0002D	TA0DRAL
0x0000E	P1PRD	0x0002E	TA0DRAH
0x0000F	P2PRD	0x0002F	TA0DRBL
0x00010	P3PRD	0x00030	TA0DRBH
0x00011	P4PRD	0x00031	TA0MOD
0x00012	P5PRD	0x00032	TA0CR
0x00013	P6PRD	0x00033	TA0SR
0x00014	P7PRD	0x00034	ADCCR1
0x00015	P8PRD	0x00035	ADCCR2
0x00016	P9PRD	0x00036	ADCDRL
0x00017	Reserved	0x00037	ADCDRH
0x00018	Reserved	0x00038	DVOCR
0x00019	Reserved	0x00039	TBTCR
0x0001A	UART0CR1	0x0003A	EIRL
0x0001B	UART0CR2	0x0003B	EIRH
0x0001C	UART0DR	0x0003C	EIRE
0x0001D	UART0SR	0x0003D	EIRD
0x0001E	TD0BUF/RD0BUF	0x0003E	Reserved
0x0001F	SIO0CR	0x0003F	PSW

Note 1: Do not access reserved addresses by the program.

Downloaded from Arrow.com.

9.2 SFR2 (0x00F00 to 0x00FFF)

Table 9-2 SFR2 (0x00F00 to 0x00F7F)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x00F00	Reserved	0x00F20	P6CR	0x00F40	Reserved	0x00F60	Reserved
0x00F01	Reserved	0x00F21	P7CR	0x00F41	Reserved	0x00F61	Reserved
0x00F02	Reserved	0x00F22	P8CR	0x00F42	Reserved	0x00F62	Reserved
0x00F03	Reserved	0x00F23	P9CR	0x00F43	P2OUTCR	0x00F63	Reserved
0x00F04	Reserved	0x00F24	Reserved	0x00F44	Reserved	0x00F64	Reserved
0x00F05	Reserved	0x00F25	Reserved	0x00F45	Reserved	0x00F65	Reserved
0x00F06	Reserved	0x00F26	Reserved	0x00F46	Reserved	0x00F66	Reserved
0x00F07	Reserved	0x00F27	P0PU	0x00F47	Reserved	0x00F67	Reserved
0x00F08	Reserved	0x00F28	P1PU	0x00F48	Reserved	0x00F68	Reserved
0x00F09	Reserved	0x00F29	P2PUD1	0x00F49	P8OUTCR	0x00F69	Reserved
0x00F0A	Reserved	0x00F2A	P3PUD1	0x00F4A	Reserved	0x00F6A	Reserved
0x00F0B	Reserved	0x00F2B	P4PU	0x00F4B	Reserved	0x00F6B	Reserved
0x00F0C	Reserved	0x00F2C	Reserved	0x00F4C	Reserved	0x00F6C	Reserved
0x00F0D	Reserved	0x00F2D	Reserved	0x00F4D	Reserved	0x00F6D	Reserved
0x00F0E	Reserved	0x00F2E	Reserved	0x00F4E	Reserved	0x00F6E	Reserved
0x00F0F	Reserved	0x00F2F	P8PU	0x00F4F	Reserved	0x00F6F	Reserved
0x00F10	Reserved	0x00F30	Reserved	0x00F50	Reserved	0x00F70	SIO1CR
0x00F11	Reserved	0x00F31	Reserved	0x00F51	Reserved	0x00F71	SIO1SR
0x00F12	Reserved	0x00F32	Reserved	0x00F52	Reserved	0x00F72	SIO1BUF
0x00F13	Reserved	0x00F33	Reserved	0x00F53	Reserved	0x00F73	Reserved
0x00F14	Reserved	0x00F34	P0FC	0x00F54	UART1CR1	0x00F74	POFFCR0
0x00F15	Reserved	0x00F35	Reserved	0x00F55	UART1CR2	0x00F75	POFFCR1
0x00F16	Reserved	0x00F36	P2FC	0x00F56	UART1DR	0x00F76	POFFCR2
0x00F17	Reserved	0x00F37	Reserved	0x00F57	UART1SR	0x00F77	POFFCR3
0x00F18	Reserved	0x00F38	P4FC	0x00F58	TD1BUF/RD1BUF	0x00F78	Reserved
0x00F19	Reserved	0x00F39	Reserved	0x00F59	Reserved	0x00F79	Reserved
0x00F1A	P0CR	0x00F3A	Reserved	0x00F5A	UART2CR1	0x00F7A	Reserved
0x00F1B	P1CR	0x00F3B	Reserved	0x00F5B	UART2CR2	0x00F7B	Reserved
0x00F1C	P2CR	0x00F3C	P8FC	0x00F5C	UART2DR	0x00F7C	SDWCR1
0x00F1D	P3CR	0x00F3D	P9FC	0x00F5D	UART2SR	0x00F7D	SDWCR2
0x00F1E	P4CR	0x00F3E	Reserved	0x00F5E	TD2BUF/RD2BUF	0x00F7E	Reserved
0x00F1F	P5CR	0x00F3F	Reserved	0x00F5F	Reserved	0x00F7F	Reserved

Note 1: Do not access reserved addresses by the program.

Downloaded from Arrow.com.

Table 9-3 SFR2 (0x00F80 to 0x00FFF)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x00F80	Reserved	0x00FA0	Reserved	0x00FC0	Reserved	0x00FE0	ILL
0x00F81	Reserved	0x00FA1	Reserved	0x00FC1	Reserved	0x00FE1	ILH
0x00F82	Reserved	0x00FA2	Reserved	0x00FC2	Reserved	0x00FE2	ILE
0x00F83	Reserved	0x00FA3	Reserved	0x00FC3	Reserved	0x00FE3	ILD
0x00F84	Reserved	0x00FA4	Reserved	0x00FC4	KWUCR0	0x00FE4	Reserved
0x00F85	Reserved	0x00FA5	Reserved	0x00FC5	Reserved	0x00FE5	Reserved
0x00F86	Reserved	0x00FA6	Reserved	0x00FC6	VDCR1	0x00FE6	Reserved
0x00F87	Reserved	0x00FA7	Reserved	0x00FC7	VDCR2	0x00FE7	Reserved
0x00F88	T02REG	0x00FA8	TB0DRAL	0x00FC8	RTCCR	0x00FE8	Reserved
0x00F89	T03REG	0x00FA9	TB0DRAH	0x00FC9	Reserved	0x00FE9	Reserved
0x00F8A	T02PWM	0x00FAA	TB0DRBL	0x00FCA	ITSEL	0x00FEA	Reserved
0x00F8B	T03PWM	0x00FAB	TB0DRBH	0x00FCB	SERSEL	0x00FEB	Reserved
0x00F8C	T02MOD	0x00FAC	TB0MOD	0x00FCC	IRSTSR	0x00FEC	Reserved
0x00F8D	T03MOD	0x00FAD	TB0CR	0x00FCD	WUCCR	0x00FED	Reserved
0x00F8E	T023CR	0x00FAE	TB0SR	0x00FCE	WUCDR	0x00FEE	Reserved
0x00F8F	Reserved	0x00FAF	Reserved	0x00FCF	CGCR	0x00FEF	Reserved
0x00F90	Reserved	0x00FB0	Reserved	0x00FD0	FLSCR1	0x00FF0	ILPRS1
0x00F91	Reserved	0x00FB1	Reserved	0x00FD1	FLSCR2/FLSCRM	0x00FF1	ILPRS2
0x00F92	Reserved	0x00FB2	Reserved	0x00FD2	Reserved	0x00FF2	ILPRS3
0x00F93	Reserved	0x00FB3	Reserved	0x00FD3	SPCR	0x00FF3	ILPRS4
0x00F94	Reserved	0x00FB4	Reserved	0x00FD4	WDCTR	0x00FF4	ILPRS5
0x00F95	Reserved	0x00FB5	Reserved	0x00FD5	WDCDR	0x00FF5	ILPRS6
0x00F96	Reserved	0x00FB6	Reserved	0x00FD6	WDCNT	0x00FF6	ILPRS7
0x00F97	Reserved	0x00FB7	Reserved	0x00FD7	WDST	0x00FF7	Reserved
0x00F98	Reserved	0x00FB8	Reserved	0x00FD8	EINTCR1	0x00FF8	Reserved
0x00F99	Reserved	0x00FB9	Reserved	0x00FD9	EINTCR2	0x00FF9	Reserved
0x00F9A	Reserved	0x00FBA	Reserved	0x00FDA	EINTCR3	0x00FFA	Reserved
0x00F9B	Reserved	0x00FBB	Reserved	0x00FDB	Reserved	0x00FFB	Reserved
0x00F9C	Reserved	0x00FBC	Reserved	0x00FDC	SYSCR1	0x00FFC	Reserved
0x00F9D	Reserved	0x00FBD	Reserved	0x00FDD	SYSCR2	0x00FFD	Reserved
0x00F9E	Reserved	0x00FBE	Reserved	0x00FDE	SYSCR3	0x00FFE	Reserved
0x00F9F	Reserved	0x00FBF	Reserved	0x00FDF	SYSCR4/SYSSR4	0x00FFF	Reserved

Note 1: Do not access reserved addresses by the program.

Downloaded from Arrow.com.

9.3 SFR3 (0x00E40 to 0x00EFF)

Table 9-4 SFR3 (0x00E40 to 0x00EBF)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x00E40	LCDBUF00	0x00E60	Reserved	0x00E80	Reserved	0x00EA0	TC0DRCH
0x00E41	LCDBUF01	0x00E61	Reserved	0x00E81	Reserved	0x00EA1	TC0DRDL
0x00E42	LCDBUF02	0x00E62	Reserved	0x00E82	Reserved	0x00EA2	TC0DRDH
0x00E43	LCDBUF03	0x00E63	Reserved	0x00E83	Reserved	0x00EA3	TC0DREL
0x00E44	LCDBUF04	0x00E64	Reserved	0x00E84	Reserved	0x00EA4	TC0DREH
0x00E45	LCDBUF05	0x00E65	Reserved	0x00E85	Reserved	0x00EA5	TC0CAPAL
0x00E46	LCDBUF06	0x00E66	Reserved	0x00E86	Reserved	0x00EA6	TC0CAPAH
0x00E47	LCDBUF07	0x00E67	Reserved	0x00E87	Reserved	0x00EA7	TC0CAPBL
0x00E48	LCDBUF08	0x00E68	Reserved	0x00E88	Reserved	0x00EA8	TC0CAPBH
0x00E49	LCDBUF09	0x00E69	Reserved	0x00E89	Reserved	0x00EA9	Reserved
0x00E4A	LCDBUF10	0x00E6A	Reserved	0x00E8A	Reserved	0x00EAA	Reserved
0x00E4B	LCDBUF11	0x00E6B	Reserved	0x00E8B	Reserved	0x00EAB	Reserved
0x00E4C	LCDBUF12	0x00E6C	Reserved	0x00E8C	Reserved	0x00EAC	Reserved
0x00E4D	LCDBUF13	0x00E6D	Reserved	0x00E8D	Reserved	0x00EAD	Reserved
0x00E4E	LCDBUF14	0x00E6E	Reserved	0x00E8E	Reserved	0x00EAE	Reserved
0x00E4F	LCDBUF15	0x00E6F	Reserved	0x00E8F	Reserved	0x00EAF	Reserved
0x00E50	LCDBUF16	0x00E70	Reserved	0x00E90	Reserved	0x00EB0	Reserved
0x00E51	LCDBUF17	0x00E71	Reserved	0x00E91	Reserved	0x00EB1	Reserved
0x00E52	LCDBUF18	0x00E72	Reserved	0x00E92	Reserved	0x00EB2	Reserved
0x00E53	LCDBUF19	0x00E73	Reserved	0x00E93	Reserved	0x00EB3	Reserved
0x00E54	Reserved	0x00E74	Reserved	0x00E94	Reserved	0x00EB4	Reserved
0x00E55	Reserved	0x00E75	Reserved	0x00E95	Reserved	0x00EB5	Reserved
0x00E56	Reserved	0x00E76	Reserved	0x00E96	Reserved	0x00EB6	Reserved
0x00E57	Reserved	0x00E77	Reserved	0x00E97	Reserved	0x00EB7	Reserved
0x00E58	Reserved	0x00E78	Reserved	0x00E98	TC0CR1	0x00EB8	Reserved
0x00E59	Reserved	0x00E79	Reserved	0x00E99	TC0CR2	0x00EB9	Reserved
0x00E5A	Reserved	0x00E7A	Reserved	0x00E9A	TC0CR3	0x00EBA	Reserved
0x00E5B	Reserved	0x00E7B	Reserved	0x00E9B	TC0DRAL	0x00EBB	Reserved
0x00E5C	Reserved	0x00E7C	LCDCR1	0x00E9C	TC0DRAH	0x00EBC	Reserved
0x00E5D	Reserved	0x00E7D	LCDCR2	0x00E9D	TC0DRBL	0x00EBD	Reserved
0x00E5E	Reserved	0x00E7E	Reserved	0x00E9E	TC0DRBH	0x00EBE	Reserved
0x00E5F	Reserved	0x00E7F	Reserved	0x00E9F	TC0DRCL	0x00EBF	Reserved

Note 1: Do not access reserved addresses by the program.

Downloaded from Arrow.com.

Table 9-5 SFR3 (0x00EC0 to 0x00EFF)

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
0x00EC0	Reserved	0x00ED0	Reserved	0x00EE0	Reserved	0x00EF0	Reserved
0x00EC1	P1LCR	0x00ED1	Reserved	0x00EE1	Reserved	0x00EF1	Reserved
0x00EC2	P2LCR	0x00ED2	P2PUD2	0x00EE2	Reserved	0x00EF2	Reserved
0x00EC3	P3LCR	0x00ED3	P3PUD2	0x00EE3	Reserved	0x00EF3	Reserved
0x00EC4	Reserved	0x00ED4	Reserved	0x00EE4	Reserved	0x00EF4	Reserved
0x00EC5	P5LCR	0x00ED5	Reserved	0x00EE5	Reserved	0x00EF5	Reserved
0x00EC6	P6LCR	0x00ED6	Reserved	0x00EE6	Reserved	0x00EF6	Reserved
0x00EC7	P7LCR	0x00ED7	Reserved	0x00EE7	Reserved	0x00EF7	Reserved
0x00EC8	Reserved	0x00ED8	Reserved	0x00EE8	Reserved	0x00EF8	Reserved
0x00EC9	Reserved	0x00ED9	Reserved	0x00EE9	Reserved	0x00EF9	Reserved
0x00ECA	Reserved	0x00EDA	Reserved	0x00EEA	Reserved	0x00EFA	Reserved
0x00ECB	Reserved	0x00EDB	Reserved	0x00EEB	Reserved	0x00EFB	Reserved
0x00ECC	Reserved	0x00EDC	Reserved	0x00EEC	Reserved	0x00EFC	Reserved
0x00ECD	Reserved	0x00EDD	Reserved	0x00EED	Reserved	0x00EFD	Reserved
0x00ECE	Reserved	0x00EDE	Reserved	0x00EEE	Reserved	0x00EFE	Reserved
0x00ECF	Reserved	0x00EDF	Reserved	0x00EEF	Reserved	0x00EFF	Reserved

Note 1: Do not access reserved addresses by the program.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

10. Low Power Consumption Function for Peripherals

The TMP89FW24A has low power consumption registers (POFFCRn) that save power when specific peripheral functions are unused. Each bit of the low power consumption registers can be set to enable or disable each peripheral function. (n = 0, 1, 2, 3)

The basic clock supply to each peripheral function is disabled for power saving, by setting the corresponding bit of the low power consumption registers (POFFCRn) to "0". (The disabled peripheral functions become unavailable.) The basic clock supply to each peripheral function is enabled and the function becomes available by setting the corresponding bit of the low power consumption registers (POFFCRn) to "1".

After reset, the low power consumption registers (POFFCRn) are initialized to "0", and thus the peripheral functions are unavailable. When each peripheral function is used for the first time, be sure to set the corresponding bit of the low power consumption registers (POFFCRn) to "1" in the initial settings of the program (before operating the control register for the peripheral function).

When a peripheral function is operating, the corresponding bit of the low power consumption registers (POFFCRn) must not be changed to "0". If it is changed, the peripheral function may operate unexpectedly.

Downloaded from Arrow.com.

10.1 Control

The low power consumption function is controlled by the low power consumption registers (POFFCRn). (n = 0, 1, 2, 3)

Low power consumption register 0

POFFCR0 (0x00F74)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCB0EN	TCA0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 Control	0 1	Disable Enable
TC001EN	TC00,01 Control	0 1	Disable Enable
TCC0EN	TCC0 Control	0 1	Disable Enable
TCB0EN	TCB0 Control	0 1	Disable Enable
TCA0EN	TCA0 Control	0 1	Disable Enable

Low power consumption register 1

POFFCR1 (0x00F75)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	SBI0EN	-	UART2EN	UART1EN	UART0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 Control	0 1	Disable Enable
UART2EN	UART2 Control	0 1	Disable Enable
UART1EN	UART1 Control	0 1	Disable Enable
UART0EN	UART0 Control	0 1	Disable Enable

Low power consumption register 2

POFFCR2 (0x00F76)	7	6	5	4	3	2	1	0
Bit Symbol	LCDEN	-	RTCEN	-	-	-	SIO1EN	SIO0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

LCDEN	LCD Control	0 1	Disable Enable
RTCEN	RTC Control	0 1	Disable Enable
SIO1EN	SIO1 Control	0 1	Disable Enable
SIO0EN	SIO0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

Low power consumption register 3

POFFCR3		7	6	5	4	3	2	1	0
(0x00F77)	Bit Symbol	-	-	INT5EN	-	INT3EN	INT2EN	INT1EN	INT0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

INT5EN	INT5 Control	0 1	Disable Enable
INT3EN	INT3 Control	0 1	Disable Enable
INT2EN	INT2 Control	0 1	Disable Enable
INT1EN	INT1 Control	0 1	Disable Enable
INT0EN	INT0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

Downloaded from Arrow.com.

11. Divider Output ($\overline{DVO}$)

This function outputs approximately 50% duty pulses that can be used to drive the piezoelectric buzzer or other device.

11.1 Configuration

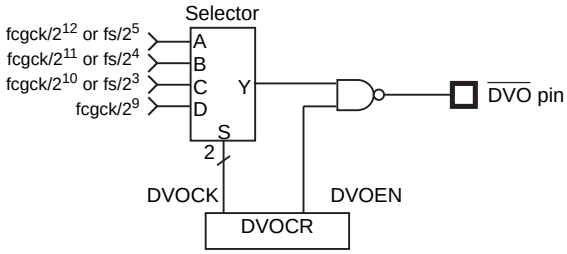


Figure 11-1 Divider Output

Downloaded from Arrow.com.

11.2 Control

The divider output is controlled by the divider output control register (DVOCR).

Divider output control register

DVOCR (0x00038)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	-	DV0EN	DVOCK	
Read/Write	R	R	R	R	R	R/W	R/W	
After reset	0	0	0	0	0	0	0	0

DVOEN	Enables/disables the divider output	0: Disable the divider output 1: Enable the divider output			
DVOCK	Selects the divider output frequency Unit: [Hz]		NORMAL 1/2, IDLE 1/2 mode		SLOW1/2 SLEEP1 mode
			DV9CK=0	DV9CK=1	
		00	fcgck/2^{12}	fs/2^5	fs/2^5
		01	fcgck/2^{11}	fs/2^4	fs/2^4
		10	fcgck/2^{10}	fs/2^3	fs/2^3
		11	fcgck/2^9	Reserved	Reserved

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: DVOCR<DVOEN> is cleared to "0" when the operation is switched to STOP or IDLE0/SLEEP0 mode. DVOCR<DVOCK> holds the value.

Note 3: When SYSCR1<DV9CK> is "1" in the NORMAL 1/2 or IDLE 1/2 mode, the DVO frequency is subject to some fluctuations to synchronize fs and fcgck.

Note 4: Bits 7 to 3 of DVOCR are read as "0".

Downloaded from Arrow.com.

11.3 Function

Select the divider output frequency at DVOCR<DVOCK>.

The divider output is enabled by setting DVOCR<DVOEN> to "1". Then, The rectangular waves selected by DVOCR<DVOCK> is output from $\overline{\text{DVO}}$ pin.

It is disabled by clearing DVOVR<DVOEN> to "0". And $\overline{\text{DVO}}$ pin keeps "H" level.

When the operation is changed to STOP or IDLE0/SLEEP0 mode, DVOCR<DVOEN> is cleared to "0" and the $\overline{\text{DVO}}$ pin outputs the "H" level.

The divider output source clock operates, regardless of the value of DVOCR<DVOEN>.

Therefore, the frequency of the first divider output after DVOCR<DVOEN> is set to "1" is not the frequency set at DVOCR<DVOCK>.

When the operation is changed to the software, STOP or IDLE0/SLEEP0 mode is activated and DVOCR<DVOEN> is cleared to "0", the frequency of the divider output is not the frequency set at DVOCR<DVOCK>.

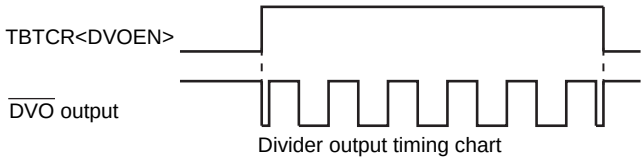


Figure 11-2 Divider Output Timing

When the operation is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, the divider output frequency does not reach the expected value due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs).

Example:2.441 kHz pulse output (fcgck = 10.0 MHz)

```
LD      (DVOCR), 0y00000100          ;DVOCK ← "00", DVOEN ← "1"
```

Table 11-1 Divider Output Frequency (Example: fcgck = 10.0 MHz, fs = 32.768 kHz)

DVOCK	Divider output frequency [Hz]		
	NORMAL 1/2, IDLE 1/2 mode		SLOW1/2, SLEEP1 mode
	DV9CK = 0	DV9CK = 1	
00	2.441 k	1.024 k	1.024 k
01	4.883 k	2.048 k	2.048 k
10	9.766 k	4.096 k	4.096 k
11	19.531 k	Reserved	Reserved

Downloaded from Arrow.com.

Downloaded from Arrow.com.

12. Time Base Timer (TBT)

The time base timer generates the time base for key scanning, dynamic display and other processes. It also provides a time base timer interrupt (INTTBT) in a certain cycle.

12.1 Time Base Timer

12.1.1 Configuration

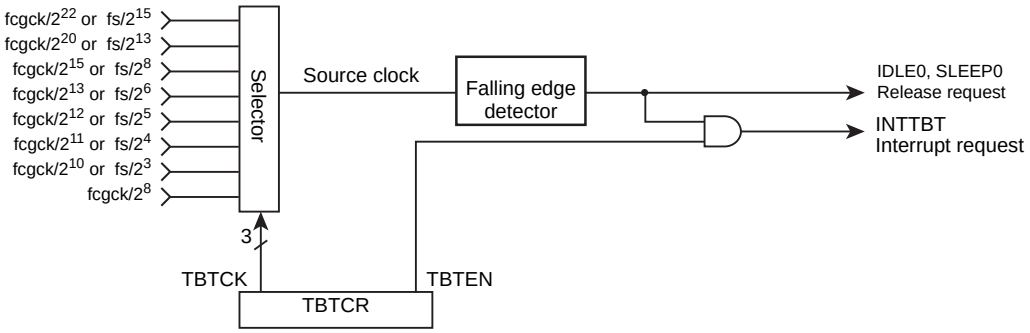


Figure 12-1 Time Base Timer Configuration

12.1.2 Control

The time base timer is controlled by the time base timer control register (TBTCR).

Time base timer control register

TBTCR (0x00039)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	TBTEN	TBTCK		
Read/Write	R	R	R	R	R/W	R/W		
After reset	0	0	0	0	0	0	0	0

TBTEN	Enables/disables the time base timer interrupt requests	0: Disables generation of interrupt request signals 1: Enables generation of interrupt request signals			
TBTCK	Selects the time base timer interrupt frequency Unit: [Hz]	TBTCK	NORMAL 1/2, IDLE 1/2 mode		SLOW1/2, SLEEP1 mode
			DV9CK = 0	DV9CK = 1	
		000	fcgck/2^{22}	fs/2^{15}	fs/2^{15}
		001	fcgck/2^{20}	fs/2^{13}	fs/2^{13}
		010	fcgck/2^{15}	fs/2^{8}	Reserved
		011	fcgck/2^{13}	fs/2^{6}	Reserved
		100	fcgck/2^{12}	fs/2^{5}	Reserved
		101	fcgck/2^{11}	fs/2^{4}	Reserved
		110	fcgck/2^{10}	fs/2^{3}	Reserved
		111	fcgck/2^{8}	Reserved	Reserved

Note 1: fcgck : Gear clock [Hz], fs : Low-frequency clock [Hz]

Note 2: When the operation is changed to the STOP mode, TBTCR<TBTEN> is cleared to "0" and TBTCR<TBTCK> maintains the value.

Note 3: TBTCR<TBTCK> should be set when TBTCR<TBTEN> is "0".

Downloaded from Arrow.com.

Note 4: When SYSCR1<DV9CK> is "1" in the NORMAL 1/2 or IDLE1/2 mode, the interrupt request is subject to some fluctuations to synchronize fs and fcgck.

Note 5: Bits 7 to 4 of TBTCR are read as "0".

12.1.3 Functions

Select the source clock frequency for the time base timer by TBTCR<TBTCK>. TBTCR<TBTCK> should be changed when TBTCR<TBTEN> is "0". Otherwise, the INTTBT interrupt request is generated at unexpected timing.

Setting TBTCR<TBTEN> to "1" causes interrupt request signals to occur at the falling edge of the source clock. When TBTCR<TBTEN> is cleared to "0", no interrupt request signal will occur.

When the operation is changed to the STOP mode, TBTCR<TBTEN> is cleared to "0".

The source clock of the time base timer operates regardless of the TBTCR<TBTEN> value.

A time base timer interrupt is generated at the first falling edge of the source clock after a time base timer interrupt request is enabled. Therefore, the period from when the time TBTCR<TBTEN> is set to "1" to the time when the first interrupt request occurs is shorter than the frequency period set at TBTCR<TBTCK>.

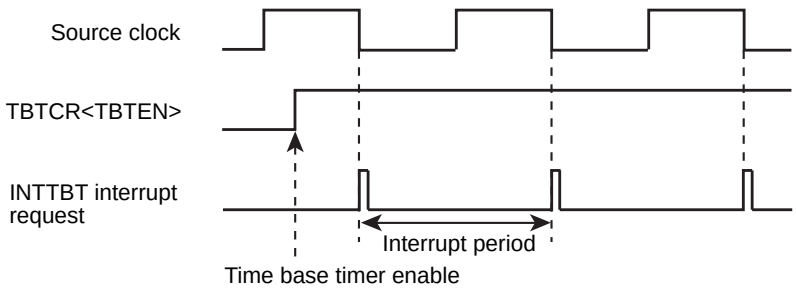


Figure 12-2 Time Base Timer Interrupt

When the operation is changed from NORMAL mode to SLOW mode or from SLOW mode to NORMAL mode, The interrupt request will not occur at the expected timing due to synchronization of the gear clock (fcgck) and the low-frequency clock (fs). It is recommened that the operation mode is changed when TBTCR<TBTEN> is "0".

Table 12-1 Time Base Timer Interrupt Frequency (Example: when fcgck = 10.0 MHz and fs = 32.768 kHz)

TBTCK	Time base timer interrupt frequency [Hz]		
	NORMAL1/2, IDLE1/2 mode	NORMAL1/2, IDLE1/2 mode	SLOW1/2, SLEEP1 mode
	DV9CK = 0	DV9CK = 1	
000	2.38	1	1
001	9.54	4	4
010	305.18	128	Reserved
011	1220.70	512	Reserved
100	2441.41	1024	Reserved
101	4882.81	2048	Reserved
110	9765.63	4096	Reserved
111	39062.5	Reserved	Reserved

Downloaded from Arrow.com.

Example:Set the time base timer interrupt frequency to fcgck/2^{15} [Hz] and enable interrupts.

```
DI                                  ;IMF ← 0
SET     (EIRL). 5                   ;Set the interrupt enable register
EI                                  ;IMF ← 1
LD      (TBTCR), 0y00000010         ;Set the interrupt frequency
LD      (TBTCR), 0y00001010         ;Enable generation of interrupt request signals
```


Downloaded from Arrow.com.

Downloaded from Arrow.com.

13. 16-bit Timer Counter (TCA)

The TMP89FW24A contains 1 channels of high-performance 16-bit timer counters (TCA).

Table 13-1 SFR Address Assignment

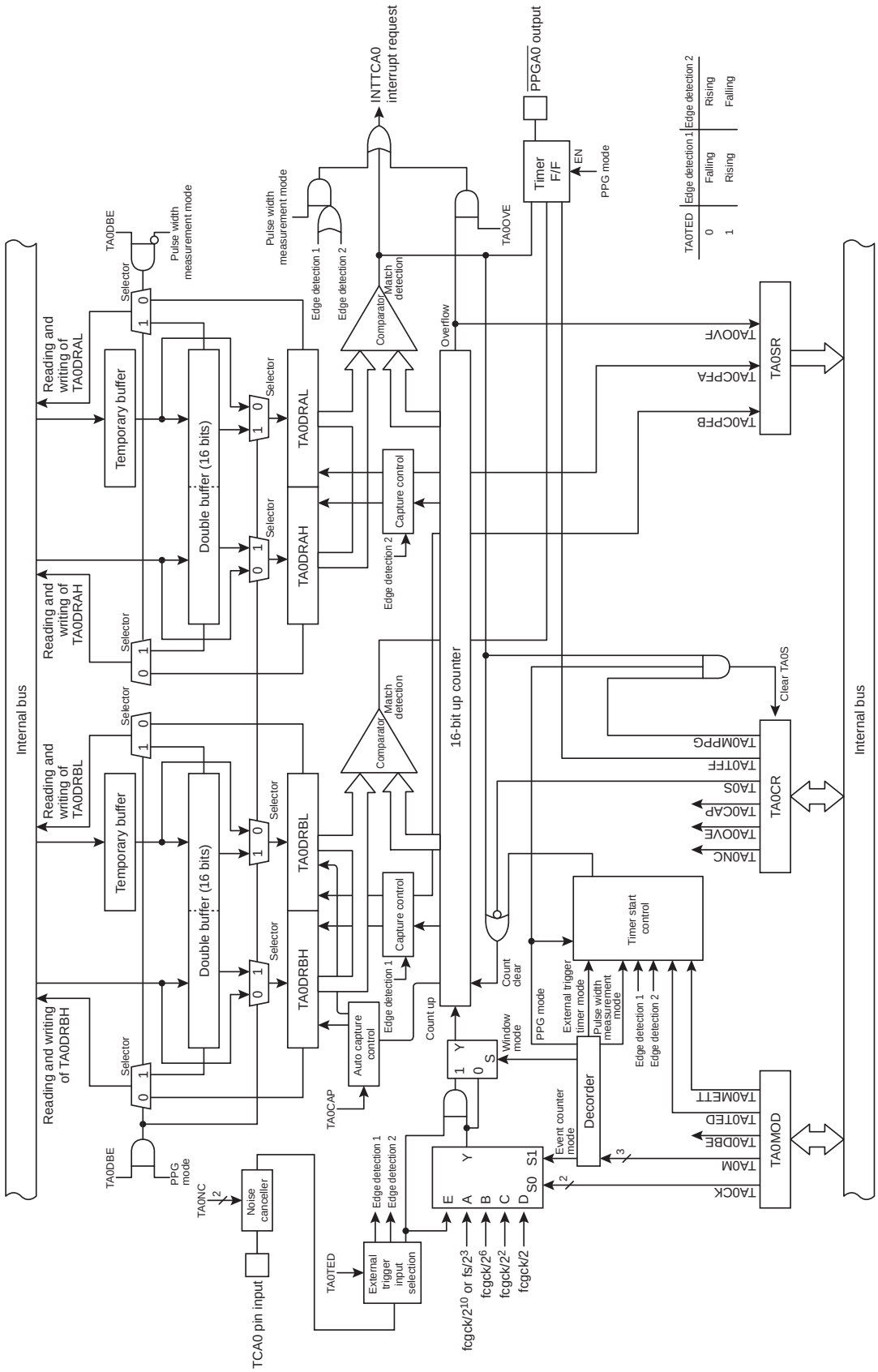
	TAxDRAL (Address)	TAxDRAH (Address)	TAxDRBL (Address)	TAxDRBH (Address)	TAxMOD (Address)	TAxCR (Address)	TAxSR (Address)	Low power consump-tion register
Timer counter A0	TA0DRAL (0x0002D)	TA0DRAH (0x0002E)	TA0DRBL (0x0002F)	TA0DRBH (0x00030)	TA0MOD (0x00031)	TA0CR (0x00032)	TA0SR (0x00033)	POFFCR0 <TCA0EN>

Table 13-2 Pin Names

	Timer input pin	PPG output pin
Timer counter A0	TCA0 pin	$\overline{\text{PPGA0}}$ pin

Downloaded from Arrow.com.

13.1 Configuration



TA0TED	Edge detection 1	Edge detection 2
0	Falling	Rising
1	Rising	Falling

Figure 13-1 Timer Counter A0

Downloaded from Arrow.com.

13.2 Control

Timer Counter A0 is controlled by the low power consumption register (POFFCR0), the timer counter A0 mode register (TA0MOD), the timer counter A0 control register (TA0CR) and two 16-bit timer A0 registers (TA0DRA and TA0DRB).

Low power consumption register 0

POFFCR0		7	6	5	4	3	2	1	0
(0x00F74)	Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCB0EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 Control	0 1	Disable Enable
TC001EN	TC00,01 Control	0 1	Disable Enable
TCC0EN	TCC0 Control	0 1	Disable Enable
TCB0EN	TCB0 Control	0 1	Disable Enable
TCA0EN	TCA0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

Timer counter A0 mode register

TA0MOD (0x00031)		7	6	5	4	3	2	1	0
	Bit Symbol	TA0DBE	TA0TED	TA0MCAP TA0METT	TA0CK		TA0M		
	Read/Write	R/W	R/W	R/W	R/W		R/W		
	After reset	1	0	0	0	0	0	0	0

TA0DBE	Double buffer control	0	Disable the double buffer		
		1	Enable the double buffer		
TA0TED	External trigger input selection	0	Rising edge/H level		
		1	Falling edge/L level		
TA0MCAP	Pulse width measurement mode control	0	Double edge capture		
		1	Single edge capture		
TA0METT	External trigger timer mode control	0	Trigger start		
		1	Trigger start & stop		
TA0CK	Timer counter 1 source clock selection		NORMAL 1/2 or IDLE 1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> ="0"	SYSCR1<DV9CK> ="1"	
		00	$fcgck/2^{10}$	$fs/2^{3}$	$fs/2^{3}$
		01	$fcgck/2^{6}$	$fcgck/2^{6}$	-
		10	$fcgck/2^{2}$	$fcgck/2^{2}$	-
		11	fcgck/2	fcgck/2	-
TA0M	Timer counter 1 operation mode selection	000	Timer mode		
		001	Timer mode		
		010	Event counter mode		
		011	PPG output mode (Software start)		
		100	External trigger timer mode		
		101	Window mode		
		110	Pulse width measurement mode		
		111			

Note 1: fcgck, Gear clock [Hz]; fs, Low-frequency clock [Hz]

Note 2: Set TA0MOD in the stopped state (TA0CR<TA0S>="0"). Writing to TA0MOD is invalid during the operation (TA0CR<TA0S>="1").

Downloaded from Arrow.com.

Timer counter A0 control register

TA0CR (0x00032)	7	6	5	4	3	2	1	0
Bit Symbol	TA0OVE	TA0TFF	TA0NC		-	-	TA0ACAP TA0MPPG	TA0S
Read/Write	R/W	R/W	R/W		R	R	R/W	R/W
After reset	0	1	0	0	0	0	0	0

TA0OVE	Overflow interrupt control	0	Generate no INTTCA0 interrupt request when the counter overflow occurs.	
		1	Generate an INTTCA0 interrupt request when the counter overflow occurs.	
TA0TFF	Timer F/F control	0	Clear	
		1	Set	
TA0NC	Noise canceller sampling interval setting		NORMAL 1/2 or IDLE 1/2 mode	SLOW1/2 or SLEEP1 mode
		00	No noise canceller	No noise canceller
		01	fcgck/2	-
		10	fcgck/2^2	-
		11	fcgck/2^8	fs/2
TA0ACAP	Auto capture function	0	Disable the auto capture	
		1	Enable the auto capture	
TA0MPPG	PPG output control	0	Continuous	
		1	One-shot	
TA0S	Timer counter A start control	0	Stop & counter clear	
		1	Start	

Note 1: The auto capture can be used only in the timer, event counter, external trigger timer and window modes.

Note 2: Set TA0TFF, TA0OVE and TA0NC in the stopped state (TA0S="0"). Writing is invalid during the operation (TA0S="1").

Note 3: When the STOP mode is started, the start control (TA0S) is automatically cleared to "0" and the timer stops. Set TA0S again to use the timer counter after the release of the STOP mode.

Note 4: When a read instruction is executed on TA0CR, bits 3 and 2 are read as "0".

Note 5: Do not set TA0NC to "01" or "10" when the SLOW 1/2 or SLEEP 1 mode is used. Setting TA0NC to "01" or "10" stops the noise canceller and no signal is input to the timer.

Downloaded from Arrow.com.

Timer counter A0 status register

TA0SR		7	6	5	4	3	2	1	0
(0x00033)	Bit Symbol	TA0OVF	-	-	-	-	-	TA0CPFA	TA0CPFB
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

TA0OVF	Overflow flag	0	No overflow has occurred.
		1	At least an overflow has occurred.
TA0CPFA	Capture completion flag A	0	No capture operation has been executed.
		1	At least a pulse width capture has been executed in the double-edge capture.
TA0CPFB	Capture completion flag B	0	No capture operation has been executed.
		1	At least a capture operation has been executed in the single-edge capture. At least a pulse duty width capture has been executed in the double-edge capture.

Note 1: TA0OVF, TA0CPFA and TA0CPFB are cleared to "0" automatically after TA0SR is read. Writing to TA0SR is invalid.
Note 2: When a read instruction is executed on TA0SR, bits 6 to 2 are read as "0".

Downloaded from Arrow.com.

Timer counter A0 register AH

TA0DRAH (0x0002E)	15	14	13	12	11	10	9	8
Bit Symbol	TA0DRAH							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer counter A0 register AL

TA0DRAL (0x0002D)	7	6	5	4	3	2	1	0
Bit Symbol	TA0DRAL							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer counter A0 register BH

TA0DRBH (0x00030)	15	14	13	12	11	10	9	8
Bit Symbol	TA0DRBH							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer counter A0 register BL

TA0DRBL (0x0002F)	7	6	5	4	3	2	1	0
Bit Symbol	TA0DRBL							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Note 1: When a write instruction is executed on TA0DRAL (TA0DRBL), the set value does not become effective immediately, but is temporarily stored in the temporary buffer. Subsequently, when a write instruction is executed on the higher-level register, TA0DRAH (TA0DRBH), the 16-bit set values are collectively stored in the double buffer or TA0DRAL/H. When setting data to the timer counter A0 register, be sure to write the data into the lower level register and the higher level in this order.

Note 2: The timer counter A0 register is not writable in the pulse width measurement mode.

Downloaded from Arrow.com.

13.3 Low Power Consumption Function

Timer counter A0 has the low power consumption register (POFFCR0) that saves power consumption when the timer is not used.

Setting POFFCR0<TCA0EN> to "0" disables the basic clock supply to timer counter A0 to save power. Note that this makes the timer unusable. Setting POFFCR0<TCA0EN> to "1" enables the basic clock supply to timer counter A0 and allows the timer to operate.

After reset, POFFCR0<TCA0EN> is initialized to "0", and this makes the timer unusable. When using the timer for the first time, be sure to set POFFCR0<TCA0EN> to "1" in the initial setting of the program (before the timer control register is operated).

Do not change POFFCR0<TCA0EN> to "0" during the timer operation. Otherwise timer counter A0 may operate unexpectedly.

Downloaded from Arrow.com.

13.4 Timer Function

Timer counter A0 has six types of operation modes; timer, external trigger timer, event counter, window, pulse width measurement and programmable pulse generate (PPG) output modes.

13.4.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock, and interrupts can be generated regularly at specified times.

13.4.1.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "000" or "001" activates the timer mode. Select the source clock at TA0MOD<TA0CK>.

Setting TA0CR<TA0S> to "1" starts the timer operation. After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> becomes invalid. Be sure to complete the required mode settings before starting the timer.

Table 13-3 Timer Mode Resolution and Maximum Time Setting

TA0MOD <TA0CK>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL 1/2 or IDLE 1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
00	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	6.7s	16s
01	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	419.4ms	-
10	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	26.2ms	-
11	fcgck/2	fcgck/2	-	200ns	-	13.1ms	-

13.4.1.2 Operation

Setting TA0CR<TA0S> to "1" allows the 16-bit up counter to increment based on the selected internal source clock. When a match between the up-counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting. Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

13.4.1.3 Auto capture

The latest contents of the up counter can be taken into timer register B (TA0DRB) by setting TA0CR<TA0ACAP> to "1" (auto capture function). When TA0CR<TA0ACAP> is "1", the current contents of the up counter can be read by reading TA0DRBL. TA0DRBH is loaded at the same time as TA0DRBL is read. Therefore, when reading the captured value, be sure to read TA0DRBL and TA0DRBH in this order. (The capture time is the timing when TA0DRBL is read.) The auto capture function can be used whether the timer is operating or stopped. When the timer is stopped, TA0DRBL is read as "0x00". TA0DRBH keeps the captured value after the timer stops, but it is cleared to "0x00" when TA0DRBL is read while the timer is stopped.

If the timer is started with TA0CR<TA0ACAP> written to "1", the auto capture is enabled immediately after the timer is started.

Note 1: The value set to TA0CR<TA0ACAP> cannot be changed at the same time as TA0CR<TA0S> is rewritten from "1" to "0". (This setting is invalid.)

Downloaded from Arrow.com.

13.4.1.4 Register buffer configuration

(1) Temporary buffer

The TMP89FW24A contains an 8-bit temporary buffer. When a write instruction is executed on TA0DRAL, the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TA0DRAH, the set value is stored into the double buffer or TA0DRAH. At the same time, the set value in the temporary buffer is stored into the double buffer or TA0DRAL. (This structure is designed to enable the set values of the lower-level and higher-level registers simultaneously.) Therefore, when setting data to TA0DRA, be sure to write the data into TA0DRAL and TA0DRAH in this order.

See Figure 13-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89FW24A, the double buffer can be used by setting TA0CR<TA0DBF>. Setting TA0CR<TA0DBF> to "0" disables the double buffer. Setting TA0CR<TA0DBF> to "1" enables the double buffer.

See Figure 13-1 for the double buffer configuration.

- When the double buffer is enabled

  When a write instruction is executed on TA0DRAH during the timer operation, the set value is first stored into the double buffer, and TA0DRAH/L are not updated immediately. TA0DRAH/L compare the up counter value to the last set values. If the values are matched, an INTTCA0 interrupt request is generated and the double buffer set value is stored in TA0DRAH/L. Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on TA0DRAH/L, the double buffer value (the last set value) is read, rather than the TA0DRAH/L values (the current effective values).

  When a write instruction is executed on TA0DRAH/L while the timer is stopped, the set value is immediately stored into both the double buffer and TA0DRAH/L.

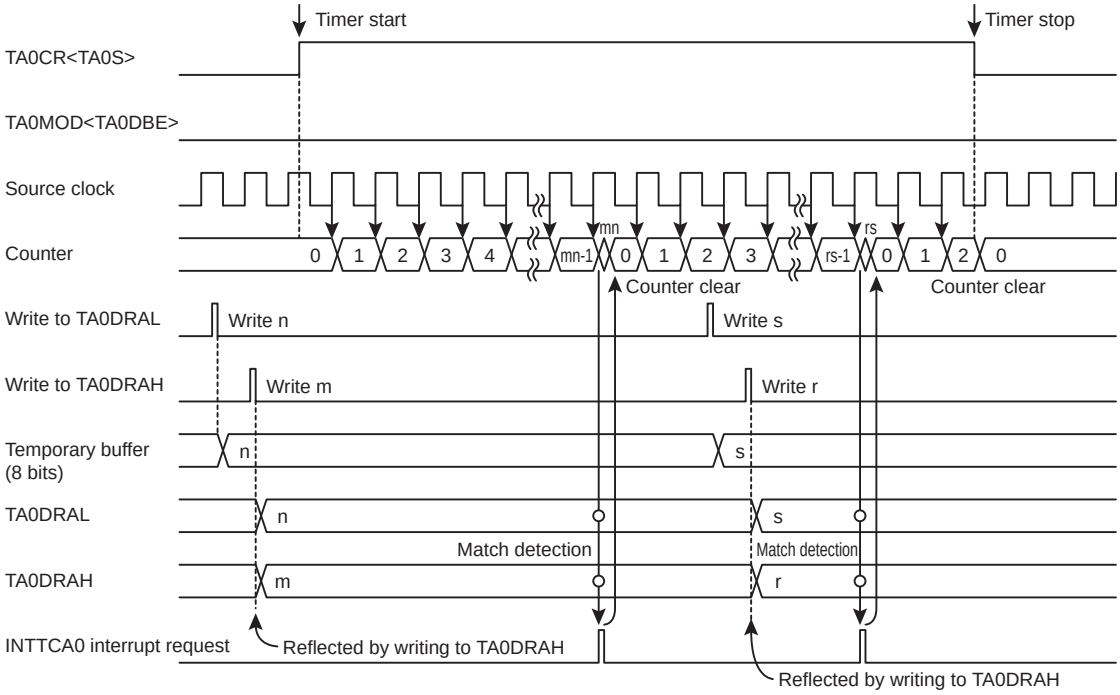
- When the double buffer is disabled

  When a write instruction is executed on TA0DRAH during the timer operation, the set value is immediately stored into TA0DRAH/L. Subsequently, the match detection is executed using a new set value.

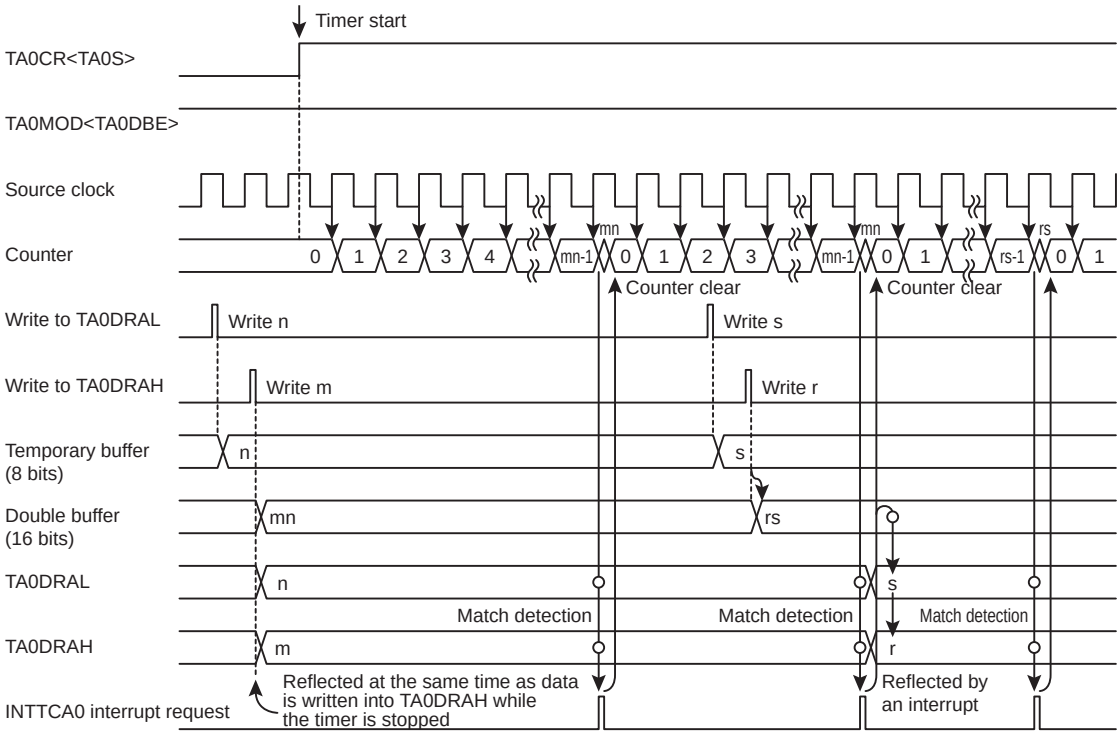
  If the values set to TA0DRAH/L are smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If that is a problem, enable the double buffer.

  When a write instruction is executed on TA0DRAH/L while the timer is stopped, the set value is immediately stored into TA0DRAH/L.

Downloaded from Arrow.com.



When the double buffer is disabled (TA0MOD<TA0DBE>="0")



When the double buffer is enabled (TA0MOD<TA0DBE>="1")

Figure 13-2 Timer Mode Timing Chart

Downloaded from Arrow.com.

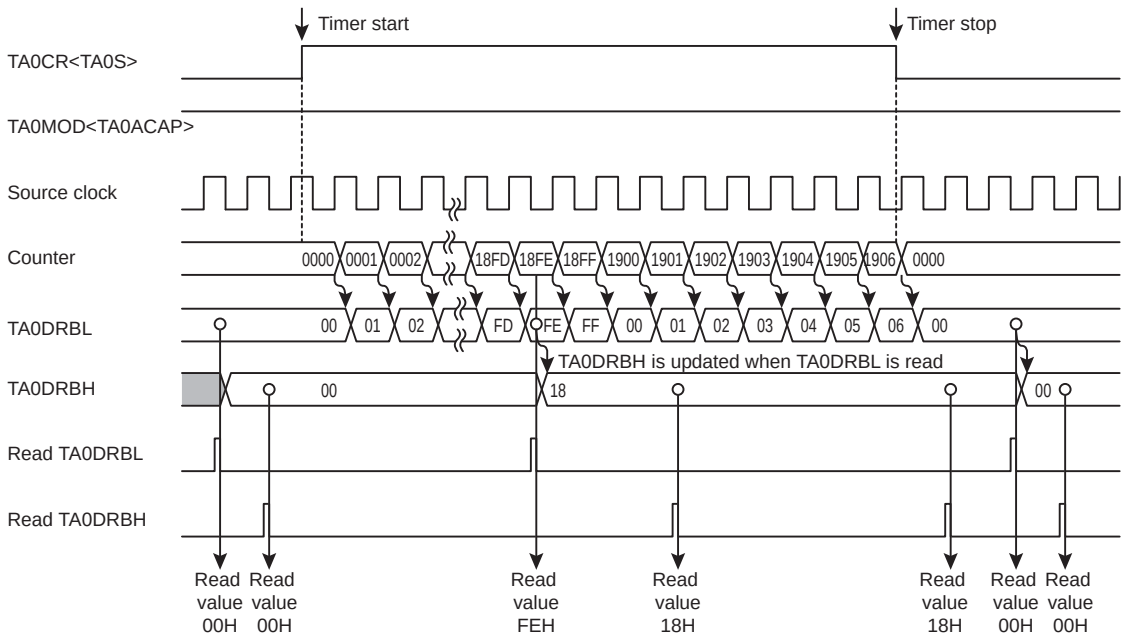


Figure 13-3 Timer Mode Timing Chart (Auto Capture)

Downloaded from Arrow.com.

13.4.2 External trigger timer mode

In the external trigger timer mode, the up counter starts counting when it is triggered by the input to the TCA0 pin.

13.4.2.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "100" activates the external trigger timer mode. Select the source clock at TA0MOD<TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.2.2 Operation

After the timer is started, when the selected trigger edge is input to the TCA0 pin, the up counter increments according to the selected source clock. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting.

When TA0MOD<TA0METT> is "1" and the edge opposite to the selected trigger edge is detected, the up counter stops counting and is cleared to "0x0000". Subsequently, when the selected trigger edge is detected, the up counter restarts counting. In this mode, an interrupt request can be generated by detecting that the input pulse exceeds a certain pulse width. If TA0MOD<TA0METT> is "0", the detection of the selected edge and the opposite edge is ignored during the period from the detection of the specified trigger edge and the start of counting through until the match detection.

Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

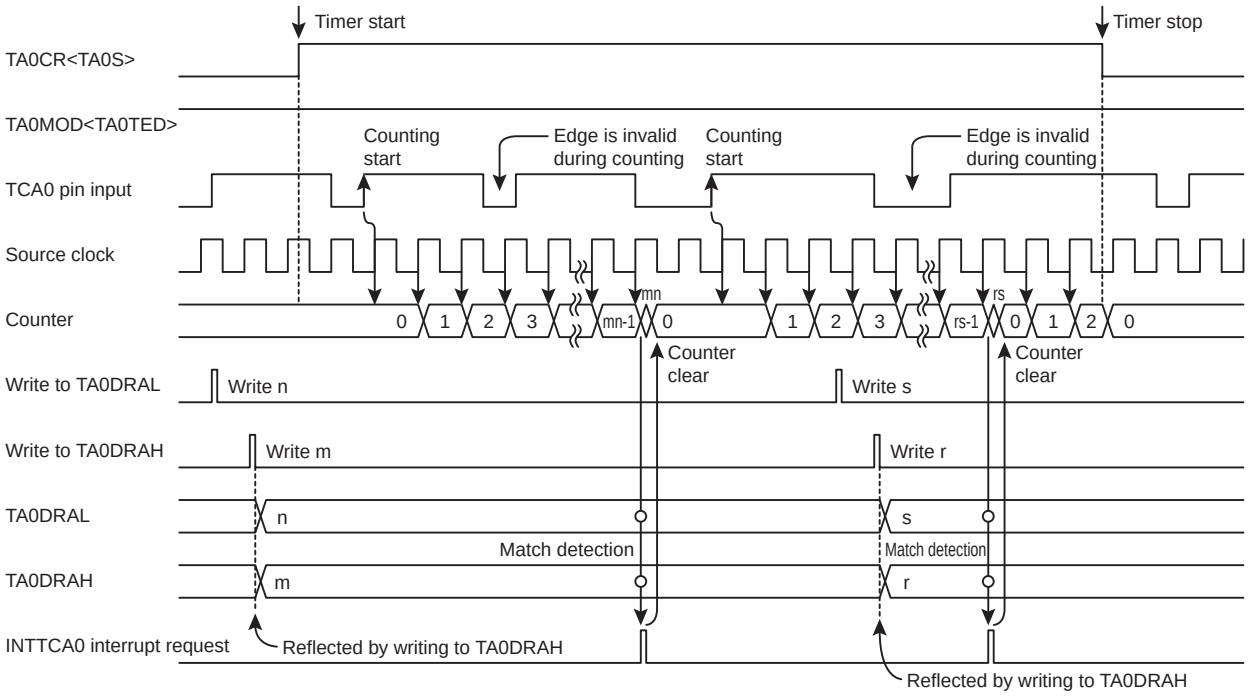
13.4.2.3 Auto capture

Refer to "13.4.1.3 Auto capture".

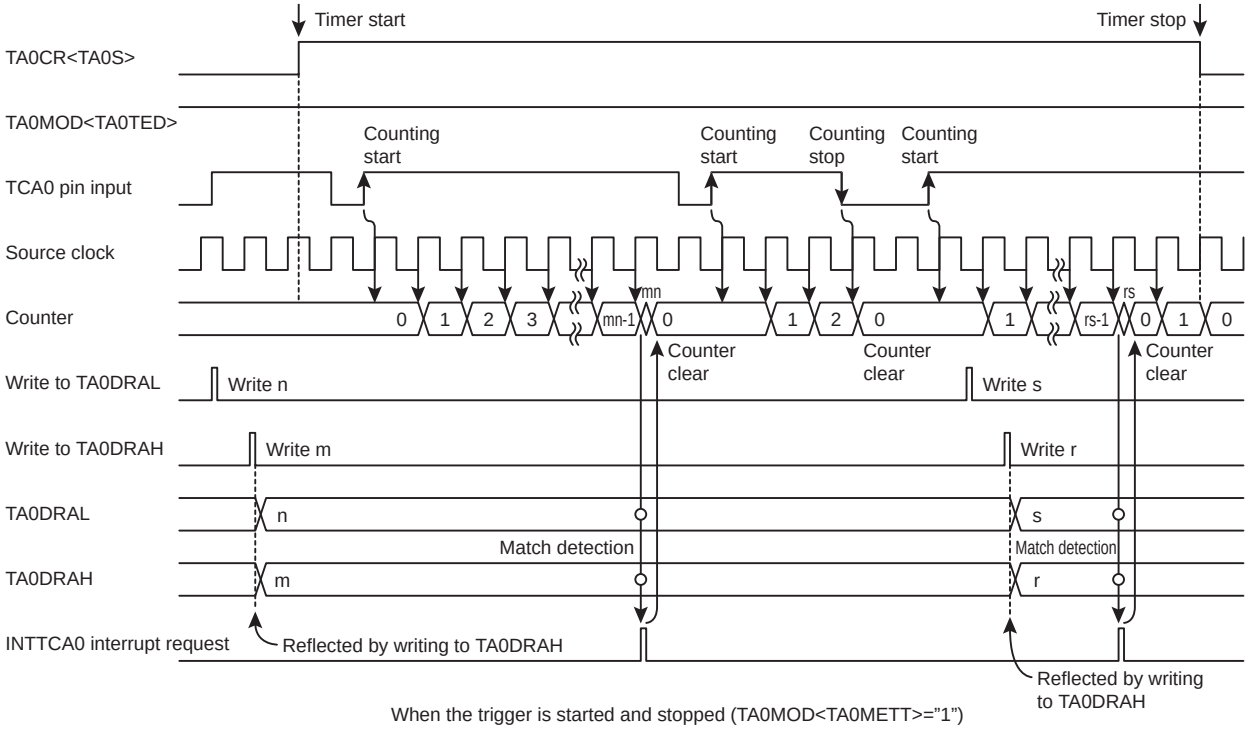
13.4.2.4 Register buffer configuration

Refer to "13.4.1.4 Register buffer configuration".

Downloaded from Arrow.com.



When the trigger is started (TA0MOD<TA0METT>="0")



When the trigger is started and stopped (TA0MOD<TA0METT>="1")

Figure 13-4 External Trigger Timer Timing Chart

Downloaded from Arrow.com.

13.4.3 Event counter mode

In the event counter mode, the up counter counts up at the edge of the input to the TCA0 pin.

13.4.3.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "010" activates the event counter mode.

Set the trigger edge at the external trigger input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge for counting up.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.3.2 Operation

After the event counter mode is started, when the selected trigger edge is input to the TCA0 pin, the up counter increments.

When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting and counts up at each edge of the input to the TCA0 pin. Setting TA0CR<TA0S> to "0" during the operation causes the up counter to stop counting and be cleared to "0x0000".

The maximum frequency to be supplied is fcgck/2 [Hz] (in the NORMAL 1/2 or IDLE 1/2 mode) or fs/2 [Hz] (in the SLOW 1/2 or SLEEP 1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

13.4.3.3 Auto capture

Refer to "13.4.1.3 Auto capture".

13.4.3.4 Register buffer configuration

Refer to "13.4.1.4 Register buffer configuration".

Downloaded from Arrow.com.

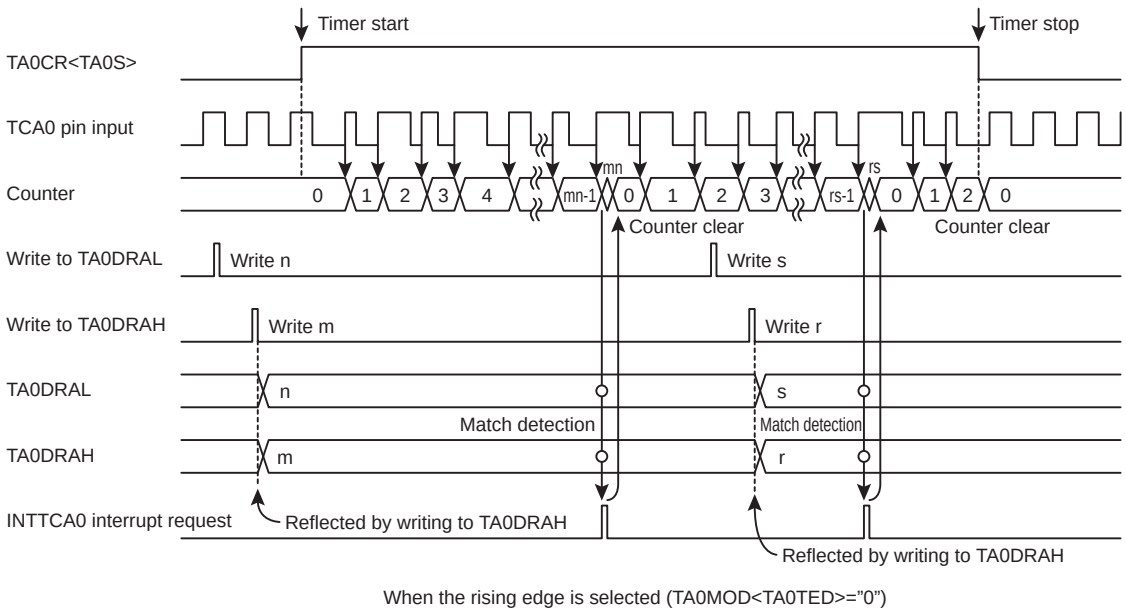


Figure 13-5 Event Count Mode Timing Chart

Downloaded from Arrow.com.

13.4.4 Window mode

In the window mode, the up counter counts up at the rising edge of the pulse that is logical anded product of the input pulse to the TCA0 pin (window pulse) and the internal clock.

13.4.4.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "101" activates the window mode. Select the source clock at TA0MOD<TA0CK>.

Select the window pulse level at the trigger edge input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" enables counting up as long as the window pulse is at the "H" level. Setting TA0MOD<TA0TED> to "1" enables counting up as long as the window pulse is at the "L" level.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.4.2 Operation

After the operation is started, when the level selected at TA0MOD<TA0TED> is input to the TCA0 pin, the up counter increments according to the source clock selected at TA0MOD<TA0CK>. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, an INTTCA0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting.

The maximum frequency to be supplied must be slow enough for the program to analyze the count value. Define a frequency pulse that is sufficiently lower than the programmed internal source clock.

Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

13.4.4.3 Auto capture

Refer to "13.4.1.3 Auto capture".

13.4.4.4 Register buffer configuration

Refer to "13.4.1.4 Register buffer configuration".

Downloaded from Arrow.com.

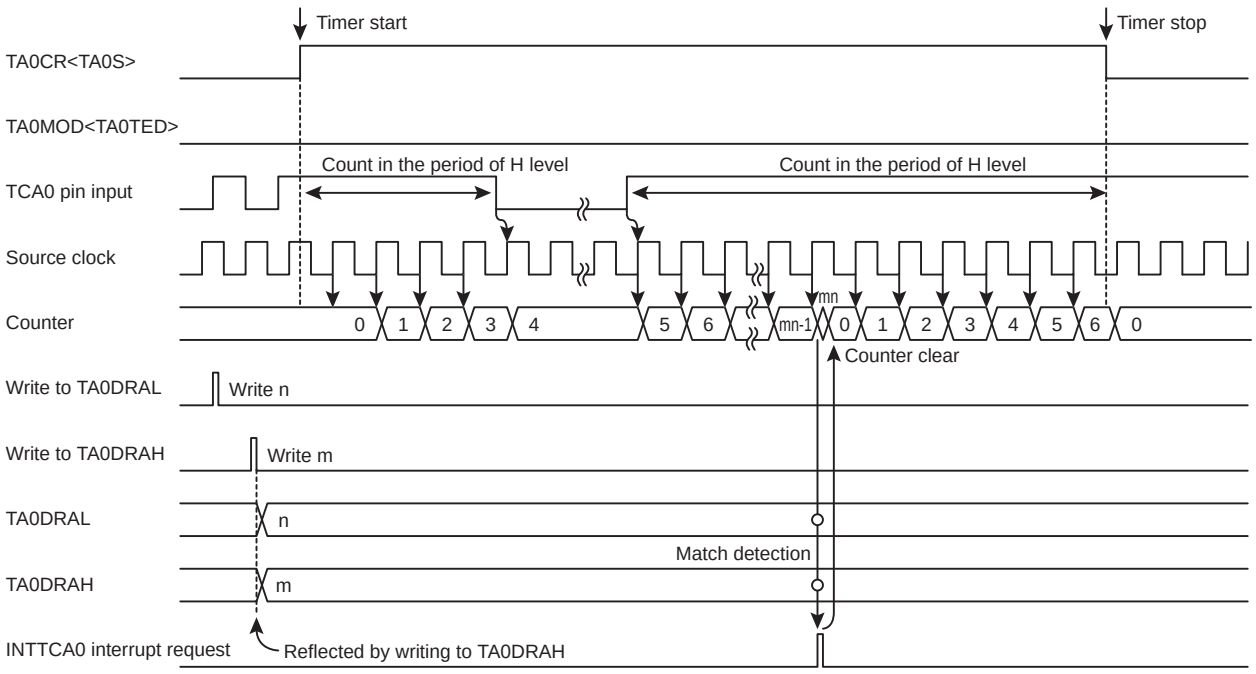


Figure 13-6 Window Mode Timing Chart

Downloaded from Arrow.com.

13.4.5 Pulse width measurement mode

In the pulse width measurement mode, the up counter starts counting at the rising/falling edge(s) of the input to the TCA0 pin and measures the input pulse width based on the internal clock.

13.4.5.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "110" activates the pulse width measurement mode. Select the source clock at TA0MOD<TA0CK>.

Select the trigger edge at the trigger edge input selection TA0MOD<TA0TED>. Setting TA0MOD<TA0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge as a trigger to start the capture.

The operation after capturing is determined by the pulse width measurement mode control TA0MOD<TA0MCAP>. Setting TA0MOD<TA0MCAP> to "0" selects the double-edge capture. Setting TA0MOD<TA0MCAP> to "1" selects the single-edge capture.

The operation to be executed in case of an overflow of the up counter can be selected at the overflow interrupt control TA0CR<TA0OVE>. Setting TA0OVE to "1" makes an INTTCA0 interrupt request occur in case of an overflow. Setting TA0OVE to "0" makes no INTTCA0 interrupt request occur in case of an overflow.

Note that this mode uses the TA0 input pin, and the TCA0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TA0CR<TA0S> to "1". In this time, TA0DRA and TA0DRB register are initialized to "0x0000". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.5.2 Operation

After the timer is started, when the selected trigger edge (start edge) is input to the TCA0 pin, INTTCA0 interrupt request is generated, and then the up counter increments according to the selected source clock. Subsequently, when the edge opposite to the selected edge is detected, the up counter value is captured into TA0DRB, an INTTCA0 interrupt request is generated, and TA0SR<TA0CPFB> is set to "1". Depending on the TA0MOD<TA0MCAP> setting, the operation differs as follows:

- Double-edge capture (When TA0MOD<TA0MCAP> is "0")

  The up counter continues counting up after the edge opposite to the selected edge is detected. Subsequently, when the selected trigger edge is input, the up counter value is captured into TA0DRA, an INTTCA0 interrupt request is generated, and TA0SR<TA0CPFA> is set to "1". At this time, the up counter is cleared to "0x0000".

- Single-edge capture (When TA0MOD<TA0MCAP> is "1")

  The up counter stops counting up and is cleared to "0x0000" when the edge opposite to the selected edge is detected. Subsequently, when the start edge is input, INTTCA0 interrupt request is generated, and then the up counter restarts increment.

When the up counter overflows during capturing, the overflow flag TA0SR<TA0OVF> is set to "1". At this time, an INTTCA0 interrupt request occurs if the overflow interrupt control TA0CR<TA0OVE> is set to "1".

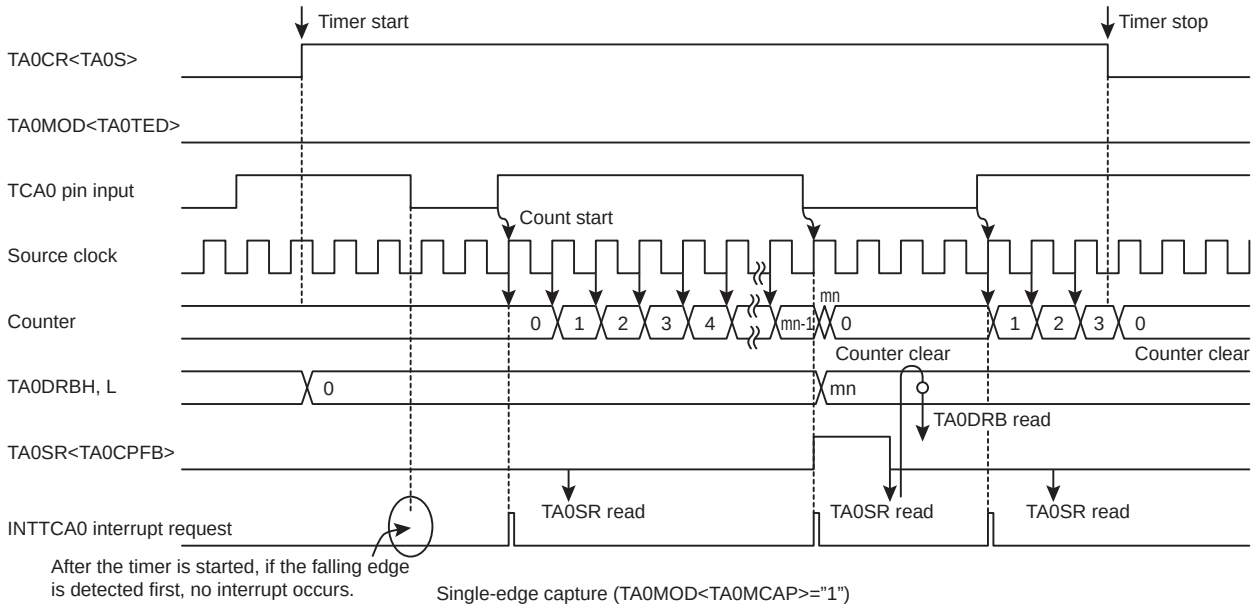
The capture completion flags (TA0SR<TA0CPFA, TA0CPFB> and the overflow flag (TA0SR<TA0OVF>) are cleared to "0" automatically when TA0SR is read.

Downloaded from Arrow.com.

The captured value must be read from TA0DRB (and also from TA0DRA for the double-edge capture) before the next trigger edge is detected. If the captured value is not read, it becomes undefined. TA0DRA and TA0DRB must be read by using a 16-bit access instruction.

Setting TA0CR<TA0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

Note 1: After the timer is started, if the edge opposite to the selected trigger edge is detected first, no capture is executed and no INTTCA0 interrupt request occurs. In this case, the capture starts when the selected trigger edge is detected next.



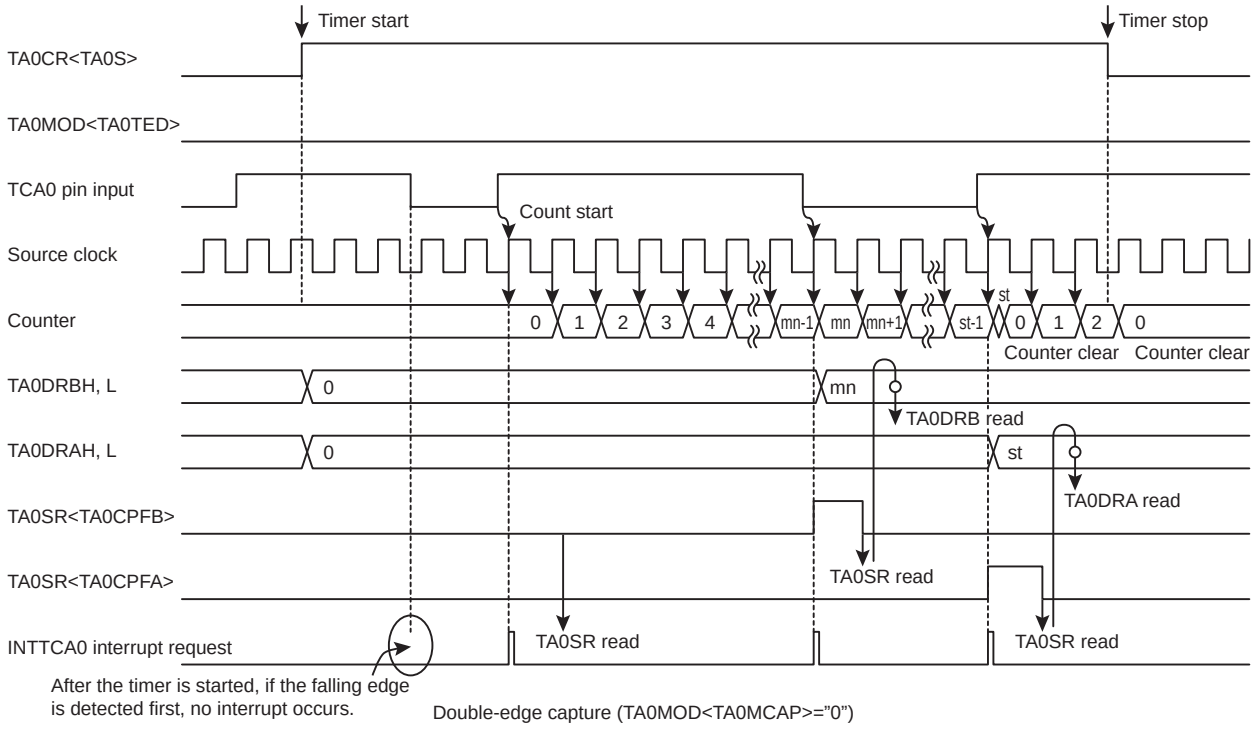


Figure 13-7 Pulse Width Measurement Mode Timing Chart

Downloaded from Arrow.com.

13.4.5.3 Capture process

Figure 13-8 shows an example of the capture process for INTTCA0 interrupt subroutine. The capture edge or overflow state can be easily judged by status register (TA0SR).

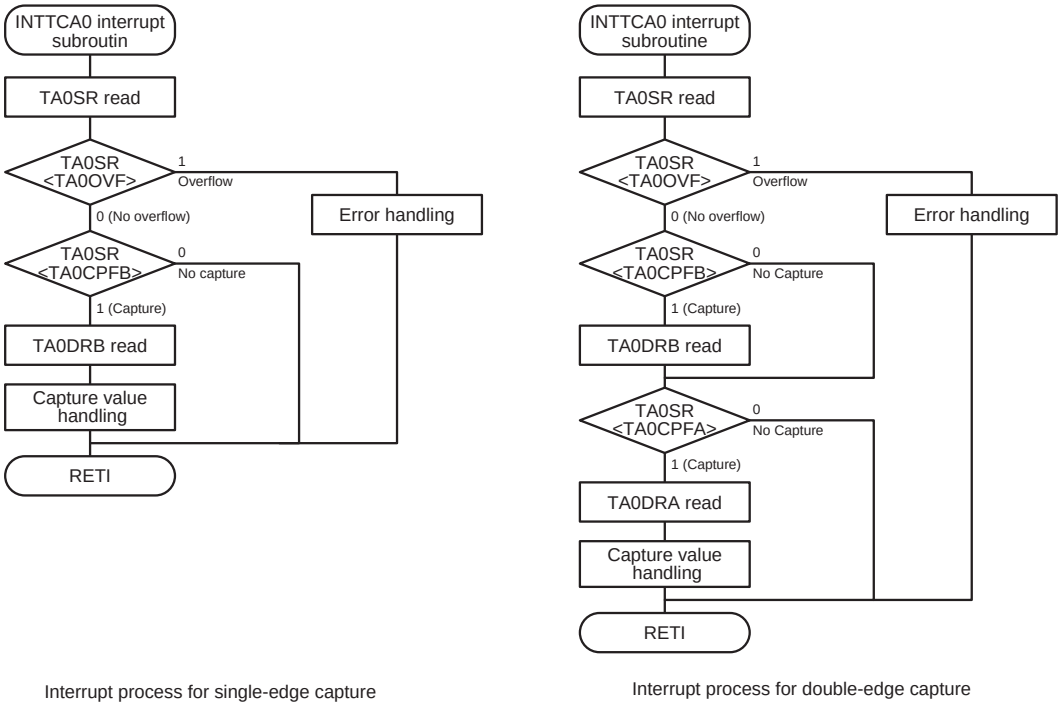


Figure 13-8 Example of capture process

Downloaded from Arrow.com.

13.4.6 Programmable pulse generate (PPG) mode

In the PPG output mode, an arbitrary duty pulse is output by two timer registers.

13.4.6.1 Setting

Setting the operation mode selection TA0MOD<TA0M> to "011" activates the PPG output mode. Select the source clock at TA0MOD<TA0CK>. Select continuous or one-shot PPG output at TA0CR<TA0MPPG>.

Set the PPG output cycle at TA0DRA and set the time until the output is reversed first at TA0DRB. Be sure to set register values so that TA0DRA is larger than TA0DRB.

Note that this mode uses the $\overline{\text{PPGA0}}$ pin. the $\overline{\text{PPGA0}}$ pin must be set to the output mode beforehand in port settings.

Set the initial state of the $\overline{\text{PPGA0}}$ pin at the timer flip-flop TA0CR<TA0TFF>. Setting TA0CR<TA0TFF> to "1" selects the "H" level as the initial state of the $\overline{\text{PPGA0}}$ pin. Setting TA0CR<TA0TFF> to "0" selects the "L" level as the initial state of the $\overline{\text{PPGA0}}$ pin.

The operation is started by setting TA0CR<TA0S> to "1". After the timer is started, writing to TA0MOD and TA0CR<TA0OVE, TA0TFF> is disabled. Be sure to complete the required mode settings before starting the timer.

13.4.6.2 Operation

after the timer is started, the up counter increments .

When a match between the up counter value and the value set to timer register B (TA0DRB) is detected, the $\overline{\text{PPGA0}}$ pin is changed to the "H" level if TA0CR<TA0TFF> is "0", or the $\overline{\text{PPGA0}}$ pin is changed to the "L" level if TA0CR<TA0TFF> is "1".

Subsequently, the up counter continues counting. When a match between the up counter value and the value set to timer register A (TA0DRA) is detected, the $\overline{\text{PPGA0}}$ pin is changed to the "L" level if TA0CR<TA0TEFF> is "0", or the $\overline{\text{PPGA0}}$ pin is changed to the "H" level if TA0CR<TA0TFF> is "1". At this time, an INTTCA0 interrupt request occurs. If the PPG output control TA0CR<TA0MPPG> is set to "1" (one-shot), TA0CR<TA0S> is automatically cleared to "0" and the timer stops.

If TA0CR<TA0MPPG> is set to "0" (continuous), the up counter is cleared to "0x0000" and continues counting and PPG output. When TA0CR<TA0S> is set to "0" (including the auto stop by the one-shot operation) during the PPG output, the $\overline{\text{PPGA0}}$ pin returns to the level set in TA0CR<TA0TFF>.

TA0CR<TA0MPPG> can be changed during the operation. Changing TA0CR<TA0MPPG> from "1" to "0" during the operation cancels the one-shot operation and enables the continuous operation. Changing TA0CR<TA0MPPG> from "0" to "1" during the operation clears TA0CR<TA0S> to "0" and stops the timer automatically after the current pulse output is completed.

Timer registers A and B can be set to the double buffer. Setting TA0CR<TA0DBF> to "1" enables the double buffer. When the values set to TA0DRA and TA0DRB are changed during the PPG output with the double buffer enabled, the writing to TA0DRA and TA0DRB will not immediately become effective but will become effective when a match between TA0DRA and the up counter is detected. If the double buffer is disabled, the writing to TA0DRA and TA0DRB will become effective immediately. If the written value is smaller than the up counter value, the up counter overflows. After a cycle, the counter match process is executed to reverse the output.

Downloaded from Arrow.com.

13.4.6.3 Register buffer configuration

(1) Temporary buffer

The TMP89FW24A contains an 8-bit temporary buffer. When a write instruction is executed on TA0DRAL (TA0DRBL), the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TA0DRAH (TA0DRBH), the set value is stored into the double buffer or TA0DRAH (TA0DRBH). At the same time, the set value in the temporary buffer is stored into the double buffer or TA0DRAL (TA0DRBL). (This structure is designed to enable the set values of the lower-level register and the higher-level register simultaneously.) Therefore, when setting data to TA0DRA (TA0DRB), be sure to write the data into TA0DRAL and TA0DRAH (TA0DRBL and TA0DRBH) in this order.

See Figure 13-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89FW24A, the double buffer can be used by setting TA0CR<TA0DBF>. Setting TA0CR<TA0DBF> to "0" disables the double buffer. Setting TA0CR<TA0DBF> to "1" enables the double buffer.

See Figure 13-1 for the double buffer configuration.

- When the double buffer is enabled

  When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is first stored into the double buffer, and TA0DRAH/L are not updated immediately. TA0DRAH/L (TA0DRBH/L) compare the last set values to the counter value. If a match is detected, an INTTCA0 interrupt request is generated and the double buffer set value is stored into TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on TA0DRAH/L (TA0DRBH/L), the double buffer value (the last set value) is read, not the TA0DRAH/L (TA0DRBH/L) values (the current effective values).

  When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into both the double buffer and TA0DRAH/L (TA0DRBH/L).

- When the double buffer is disabled

  When a write instruction is executed on TA0DRAH (TA0DRBH) during the timer operation, the set value is immediately stored in TA0DRAH/L (TA0DRBH/L). Subsequently, the match detection is executed using a new set value.

  If the values set to TA0DRAH/L (TA0DRBH/L) are smaller than the up counter value, the up counter overflows and the match detection is executed using a new set value. As a result, the output pulse width may be longer than the set time. If that is a problem, enable the double buffer.

  When a write instruction is executed on TA0DRAH/L (TA0DRBH/L) while the timer is stopped, the set value is immediately stored into TA0DRAH/L (TA0DRBH/L).

Downloaded from Arrow.com.

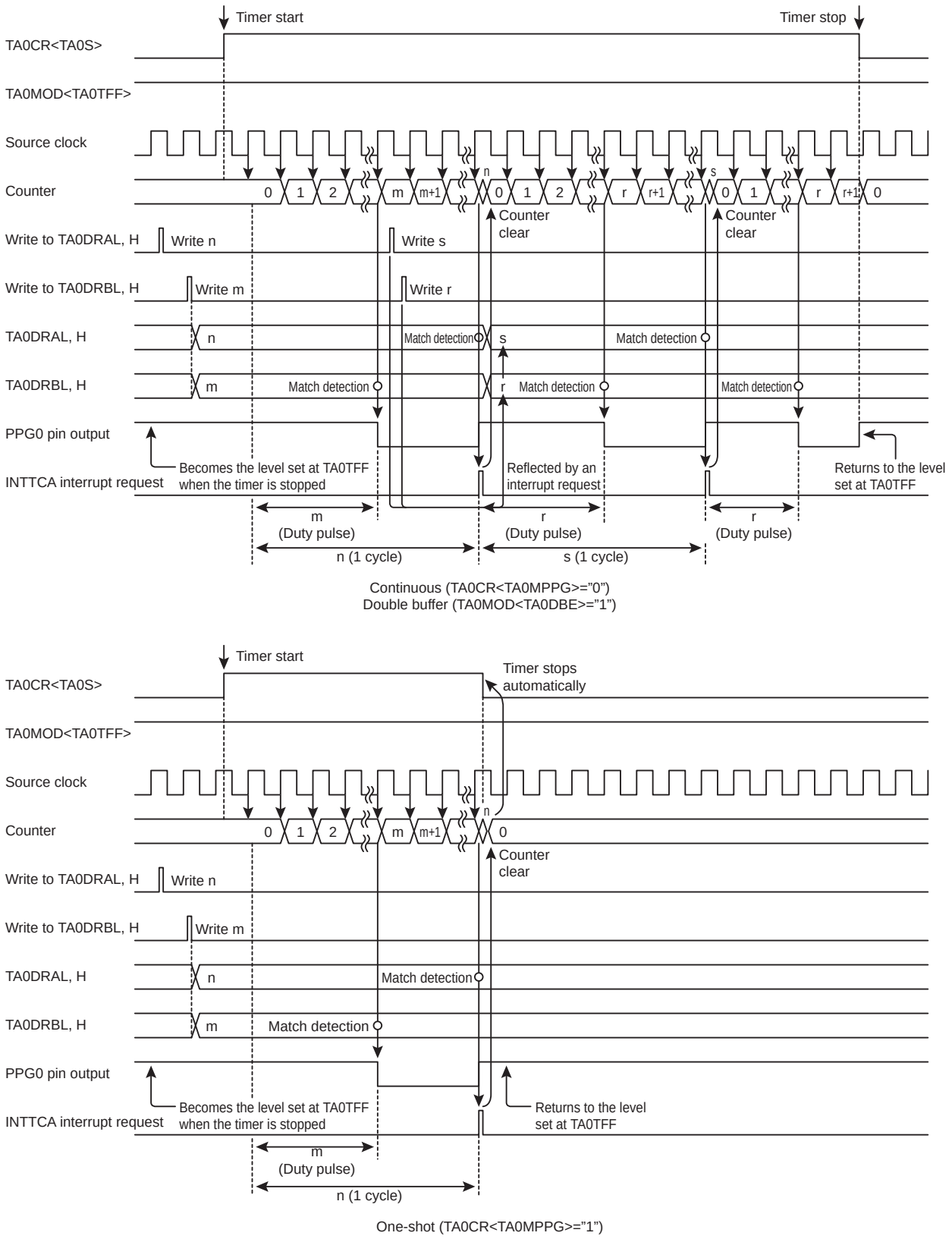


Figure 13-9 PPG Mode Timing Chart

Downloaded from Arrow.com.

13.5 Noise Canceller

The digital noise canceller can be used in the operation modes that use the TCA0 pin.

13.5.1 Setting

When the digital noise canceller is used, the input level is sampled at the sampling intervals set at TA0CR<TA0NC>. When the same level is detected three times consecutively, the level of the input to the timer is changed.

Setting TA0CR<TA0NC> to any values than "00" allows the noise canceller to start operation, regardless of the TA0CR<TA0S> value.

When the noise canceller is used, allow the timer to start after a period of time that is equal to four times the sampling interval after TA0CR<TA0NC> is set has elapsed. This stabilizes the input signal.

Set TA0CR<TA0NC> while the timer is stopped (TA0CR<TA0S> = "0"). When TA0CR<TA0S> is "1", writing is ignored.

In the SLOW 1/2 or SLEEP 1 mode, setting TA0CR<TA0NC> to "11" selects fs/2 as the source clock for the operation. Setting TA0CR<TA0NC> to "00" disables the noise canceller. Setting TA0CR<TA0NC> to "01" or "10" disables the TCA0 pin input.

Table 13-4 Noise Cancel Time (fcgck = 10 [MHz])

TA0NC	Sampling interval	Time removed as noise	Time regarded as signal
00	None	-	-
01	200 ns (2/fcgck)	600 ns or less	800 ns or more
10	400 ns (4/fcgck)	1.2 μs or less	1.6 μs or more
11	25.6 μs (256/fcgck)	76.8 μs or less	102.4 μs or more

Downloaded from Arrow.com.

Downloaded from Arrow.com.

14. 16-bit Timer Counter (TCB)

The TMP89FW24A contains one channel of high-performance 16-bit timer counter (TCB0).

Table 14-1 SFR Address Assignment

	TBxDRAL (Address)	TBxDRAH (Address)	TBxDRBL (Address)	TBxDRBH (Address)	TBxMOD (Address)	TBxCR (Address)	TBxSR (Address)	Low power consump-tion register
Timer counter B0	TB0DRAL (0x00FA8)	TB0DRAH (0x00FA9)	TB0DRBL (0x00FAA)	TB0DRBH (0x00FAB)	TB0MOD (0x00FAC)	TB0CR (0x00FAD)	TB0SR (0x00FAE)	POFFCR0 <TCB0EN>

Table 14-2 Pin Names

	Timer input pin	PPG output pin
Timer counter B0	TCB0 pin	$\overline{\text{PPGB0}}$ pin

Downloaded from Arrow.com.

14.1 Configuration

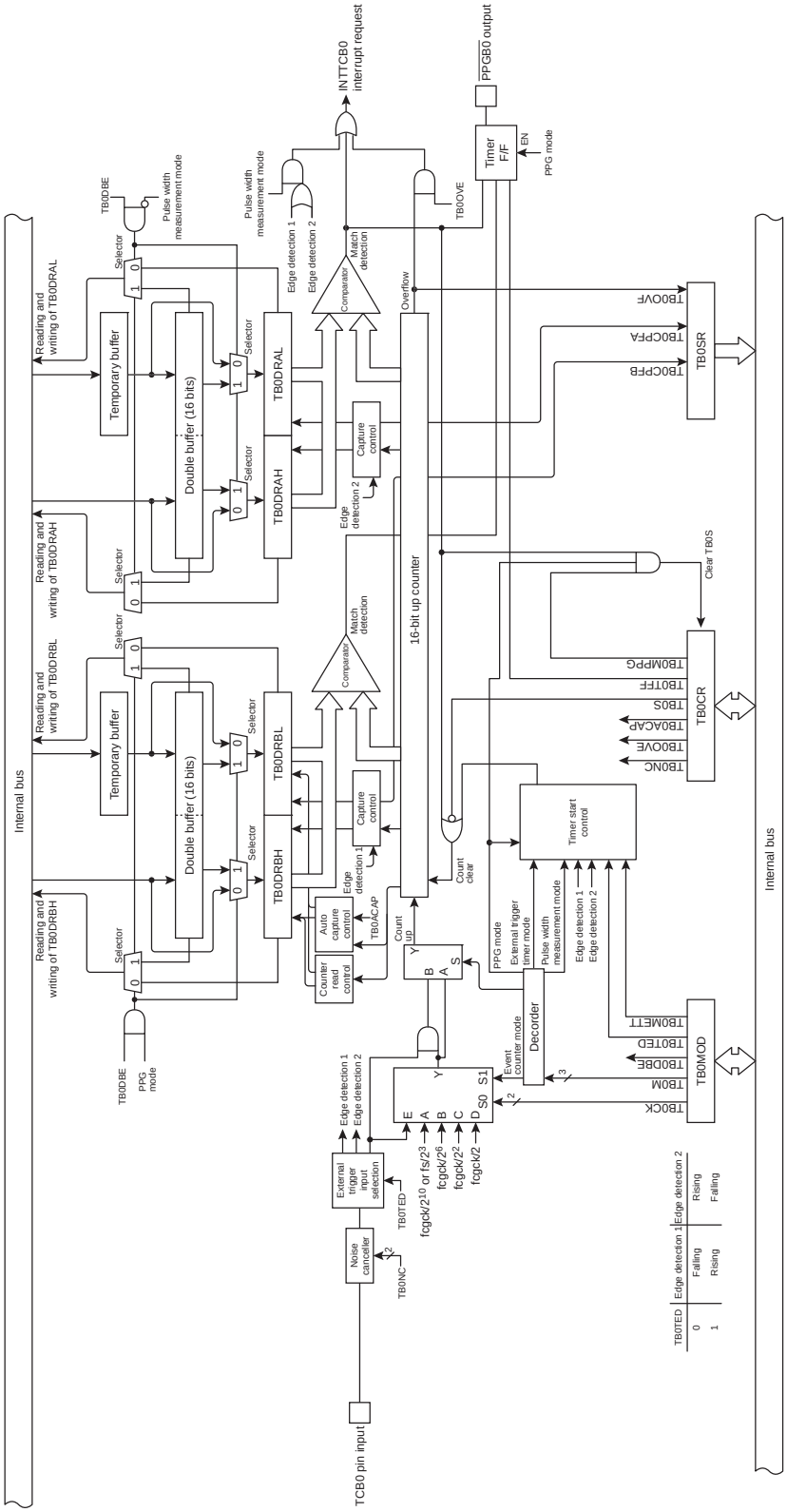


Figure 14-1 Timer Counter B0

Downloaded from Arrow.com.

14.2 Control

Timer Counter B0 is controlled by the low power consumption register (POFFCR01234), the timer counter B0 mode register (TB0MOD), the timer counter B0 control register (TB0CR) and two 16-bit timer B0 registers (TB0DRA and TB0DRB).

Low power consumption register 0

POFFCR0		7	6	5	4	3	2	1	0
(0x00F74)	Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCB0EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 Control	0 1	Disable Enable
TC001EN	TC00,01 Control	0 1	Disable Enable
TCC0EN	TCC0 Control	0 1	Disable Enable
TCB0EN	TCB0 Control	0 1	Disable Enable
TCA0EN	TCA0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

Timer counter B0 mode register

TB0MOD (0x00FAC)		7	6	5	4	3	2	1	0
	Bit Symbol	TB0DBE	TB0TED	TB0MCAP TB0METT	TB0CK		TB0M		
	Read/Write	R/W	R/W	R/W	R/W		R/W		
	After reset	1	0	0	0	0	0	0	0

TB0DBE	Double buffer control (TB0DRA and TB0DRB registers)	0 1	Disable the double buffer Enable the double buffer		
TB0TED	External trigger input selection	0 1	Rising edge/H level Falling edge/L level		
TB0MCAP	Pulse width measurement mode control	0 1	Double edge capture Single edge capture		
TB0METT	External trigger timer mode control	0 1	Trigger start Trigger start & stop		
TB0CK	Timer counter 1 source clock selection		NORMAL 1/2 or IDLE 1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> ="0"	SYSCR1<DV9CK> ="1"	
		00	fcgck/2^{10}	fs/2^3	fs/2^3
		01	fcgck/2^6	fcgck/2^6	-
		10	fcgck/2^2	fcgck/2^2	-
		11	fcgck/2	fcgck/2	-
TB0M	Timer counter 1 operation mode selection	000	Timer mode		
		001	Reserved		
		010	Event counter mode		
		011	PPG output mode (Software start)		
		100	External trigger timer mode		
		101	Window mode		
		110	Pulse width measurement mode		
		111	PPG output mode (External trigger start)		

Note 1: fcgck, Gear clock [Hz]; fs, Low-frequency clock [Hz]

Note 2: Set TB0MOD in the stopped state (TB0CR<TB0S>="0"). Writing to TB0MOD is invalid during the operation (TB0CR<TB0S>="1").

Downloaded from Arrow.com.

Timer counter B0 control register

TB0CR (0x00FAD)	7	6	5	4	3	2	1	0
Bit Symbol	TB0OVE	TB0TFF	TB0NC		-	-	TB0ACAP TB0MPPG	TB0S
Read/Write	R/W	R/W	R/W		R	W	R/W	R/W
After reset	0	1	0	0	0	0	0	0

TB0OVE	Overflow interrupt control	0	Generate no INTTCB0 interrupt request when the counter overflow occurs.	
		1	Generate an INTTCB0 interrupt request when the counter overflow occurs.	
TB0TFF	Timer F/F control	0	Clear	
		1	Set	
TB0NC	Noise canceller sampling interval setting		NORMAL 1/2 or IDLE 1/2 mode	SLOW1/2 or SLEEP1 mode
		00	No noise canceller	No noise canceller
		01	fcgck/2	-
		10	fcgck/2^2	-
		11	fcgck/2^8	fs/2
TB0ACAP	Auto capture function	0	Disable the auto capture	
		1	Enable the auto capture	
TB0MPPG	PPG output control	0	Continuous	
		1	One-shot	
TB0S	Timer counter A start control	0	Stop & counter clear	
		1	Start	

Note 1: The auto capture can be used only in the timer, event counter, external trigger timer and window modes.

Note 2: Set TB0TFF, TB0OVE and TB0NC in the stopped state (TB0S="0"). Writing is invalid during the operation (TB0S="1").

Note 3: When the STOP mode is started, the start control (TB0S) is automatically cleared to "0" and the timer stops. Set TB0S again to use the timer counter after the release of the STOP mode.

Note 4: When a read instruction is executed on TB0CR, bits 3 and 2 are read as "0".

Note 5: Do not set TB0NC to "01" or "10" when the SLOW 1/2 or SLEEP 1 mode is used. Setting TB0NC to "01" or "10" stops the noise canceller and no signal is input to the timer.

Downloaded from Arrow.com.

Timer counter B0 status register

TB0SR (0x00FAE)	7	6	5	4	3	2	1	0
Bit Symbol	TB0OVF	-	-	-	-	-	TB0CPFA	TB0CPFB
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

TB0OVF	Overflow flag	0	No overflow has occurred.
		1	At least an overflow has occurred.
TB0CPFA	Capture completion flag A	0	No capture operation has been executed.
		1	At least a pulse width capture has been executed in the double-edge capture.
TB0CPFB	Capture completion flag B	0	No capture operation has been executed.
		1	At least a capture operation has been executed in the single-edge capture. At least a pulse duty width capture has been executed in the double-edge capture.

Note 1: TB0OVF, TB0CPFA and TB0CPFB are cleared to "0" automatically after TB0SR is read. Writing to TB0SR is invalid.
Note 2: When a read instruction is executed on TB0SR, bits 6 to 3 are read as "0".

Timer counter B0 register AH

TB0DRAH (0x00FA9)	15	14	13	12	11	10	9	8
Bit Symbol	TB0DRAH							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer counter B0 register AL

TB0DRAL (0x00FA8)	7	6	5	4	3	2	1	0
Bit Symbol	TB0DRAL							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer counter B0 register BH

TB0DRBH (0x00FAB)	15	14	13	12	11	10	9	8
Bit Symbol	TB0DRBH							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer counter B0 register BL

TB0DRBL (0x00FAA)	7	6	5	4	3	2	1	0
Bit Symbol	TB0DRBL							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Note 1: When a write instruction is executed on TB0DRAL (TB0DRBL), the set value does not become effective immediately, but is temporarily stored in the temporary buffer. Subsequently, when a write instruction is executed on the higher-level register, TB0DRAH (TB0DRBH), the 16-bit set values are collectively stored in the double buffer or TB0DRAL/H. When setting data to the timer counter B0 register, be sure to write the data into the lower level register and the higher level in this order.
Note 2: The timer counter B0 register is not writable in the pulse width measurement mode.

Downloaded from Arrow.com.

14.3 Low Power Consumption Function

Timer counter B0 has the low power consumption register (POFFCR0) that saves power consumption when the timer is not used.

Setting POFFCR0<TCB0EN> to "0" disables the basic clock supply to timer counter B0 to save power. Note that this makes the timer unusable. Setting POFFCR0<TCB0EN> to "1" enables the basic clock supply to timer counter B0 and allows the timer to operate.

After reset, POFFCR0<TCB0EN> is initialized to "0", and this makes the timer unusable. When using the timer for the first time, be sure to set POFFCR0<TCB0EN> to "1" in the initial setting of the program (before the timer control register is operated).

Do not change POFFCR0<TCB0EN> to "0" during the timer operation. Otherwise timer counter B0 may operate unexpectedly.

Downloaded from Arrow.com.

14.4 Timer Function

Timer counter B0 has six types of operation modes; timer, external trigger timer, event counter, window, pulse width measurement and programmable pulse generate (PPG) output modes.

14.4.1 Timer mode

In the timer mode, the up-counter counts up using the internal clock, and interrupts can be generated regularly at specified times.

14.4.1.1 Setting

Setting the operation mode selection TB0MOD<TB0M> to "000" or "001" activates the timer mode. Select the source clock at TB0MOD<TB0CK>.

Setting TB0CR<TB0S> to "1" starts the timer operation. After the timer is started, writing to TB0MOD and TB0CR<TB0OVE> becomes invalid. Be sure to complete the required mode settings before starting the timer.

Table 14-3 Timer Mode Resolution and Maximum Time Setting

TB0MOD <TB0CK>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL 1/2 or IDLE 1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
00	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	6.7s	16s
01	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	419.4ms	-
10	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	26.2ms	-
11	fcgck/2	fcgck/2	-	200ns	-	13.1ms	-

14.4.1.2 Operation

Setting TB0CR<TB0S> to "1" allows the 16-bit up counter to increment based on the selected internal source clock. When a match between the up-counter value and the value set to timer register A (TB0DRA) is detected, an INTTCB0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting. Setting TB0CR<TB0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

14.4.1.3 Auto capture

The latest contents of the up counter can be taken into timer register B (TB0DRB) by setting TB0CR<TB0ACAP> to "1" (auto capture function). When TB0CR<TB0ACAP> is "1", the current contents of the up counter can be read by reading TB0DRBL. TB0DRBH is loaded at the same time as TB0DRBL is read. Therefore, when reading the captured value, be sure to read TB0DRBL and TB0DRBH in this order. (The capture time is the timing when TB0DRBL is read.) The auto capture function can be used whether the timer is operating or stopped. When the timer is stopped, TB0DRBL is read as "0x00". TB0DRBH keeps the captured value after the timer stops, but it is cleared to "0x00" when TB0DRBL is read while the timer is stopped.

If the timer is started with TB0CR<TB0ACAP> written to "1", the auto capture is enabled immediately after the timer is started.

Note 1: The value set to TB0CR<TB0ACAP> cannot be changed at the same time as TB0CR<TB0S> is rewritten from "1" to "0". (This setting is invalid.)

Downloaded from Arrow.com.

14.4.1.4 Register buffer configuration

(1) Temporary buffer

The TMP89FW24A contains an 8-bit temporary buffer. When a write instruction is executed on TB0DRAL, the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TB0DRAH, the set value is stored into the double buffer or TB0DRAH. At the same time, the set value in the temporary buffer is stored into the double buffer or TB0DRAL. (This structure is designed to enable the set values of the lower-level and higher-level registers simultaneously.) Therefore, when setting data to TB0DRA, be sure to write the data into TB0DRAL and TB0DRAH in this order.

See Figure 14-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89FW24A, the double buffer can be used by setting TB0CR<TB0DBF>. Setting TB0CR<TB0DBF> to "0" disables the double buffer. Setting TB0CR<TB0DBF> to "1" enables the double buffer.

See Figure 14-1 for the double buffer configuration.

- When the double buffer is enabled

  When a write instruction is executed on TB0DRAH during the timer operation, the set value is first stored into the double buffer, and TB0DRAH/L are not updated immediately. TB0DRAH/L compare the up counter value to the last set values. If the values are matched, an INTTCB0 interrupt request is generated and the double buffer set value is stored in TB0DRAH/L. Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on TB0DRAH/L, the double buffer value (the last set value) is read, rather than the TB0DRAH/L values (the current effective values).

  When a write instruction is executed on TB0DRAH/L while the timer is stopped, the set value is immediately stored into both the double buffer and TB0DRAH/L.

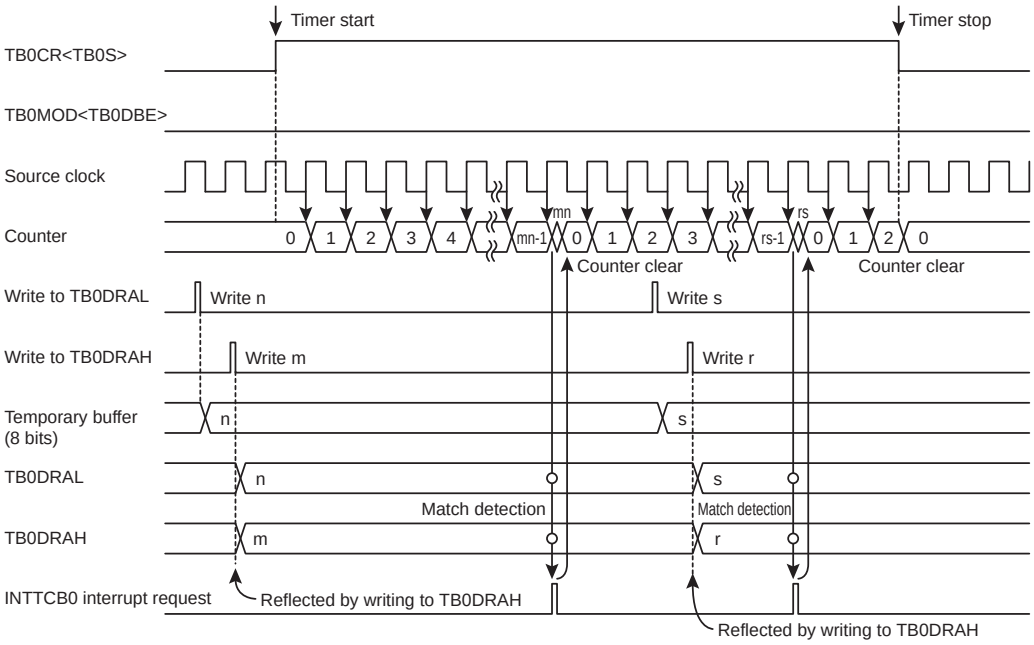
- When the double buffer is disabled

  When a write instruction is executed on TB0DRAH during the timer operation, the set value is immediately stored into TB0DRAH/L. Subsequently, the match detection is executed using a new set value.

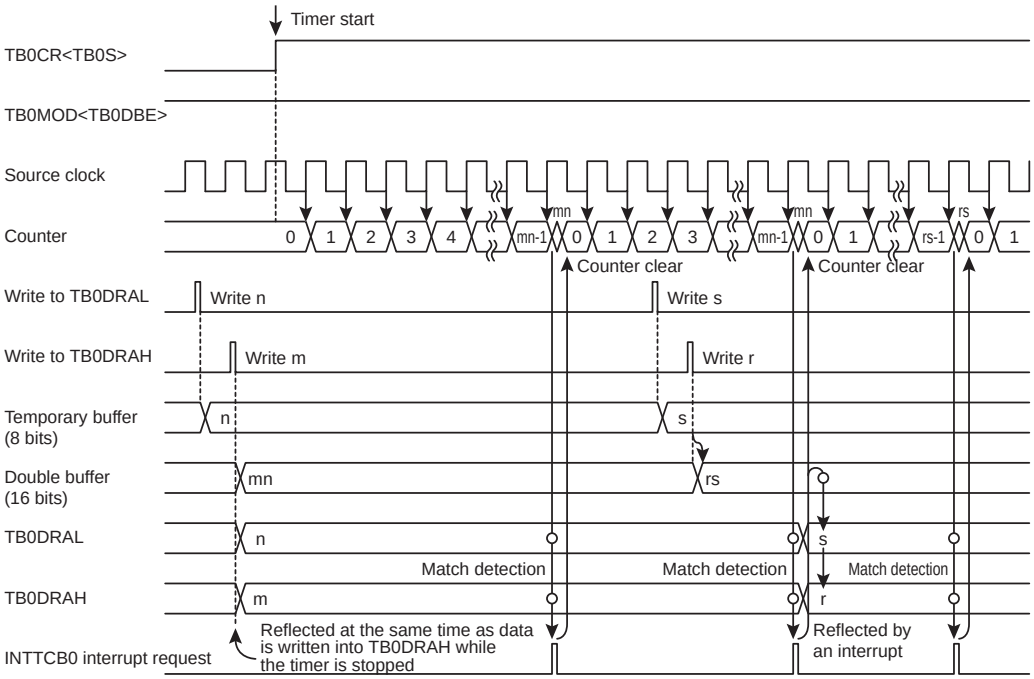
  If the values set to TB0DRAH/L are smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If that is a problem, enable the double buffer.

  When a write instruction is executed on TB0DRAH/L while the timer is stopped, the set value is immediately stored into TB0DRAH/L.

Downloaded from Arrow.com.



When the double buffer is disabled (TB0MOD<TB0DBE>="0")



When the double buffer is enabled (TB0MOD<TB0DBE>="1")

Figure 14-2 Timer Mode Timing Chart

Downloaded from Arrow.com.

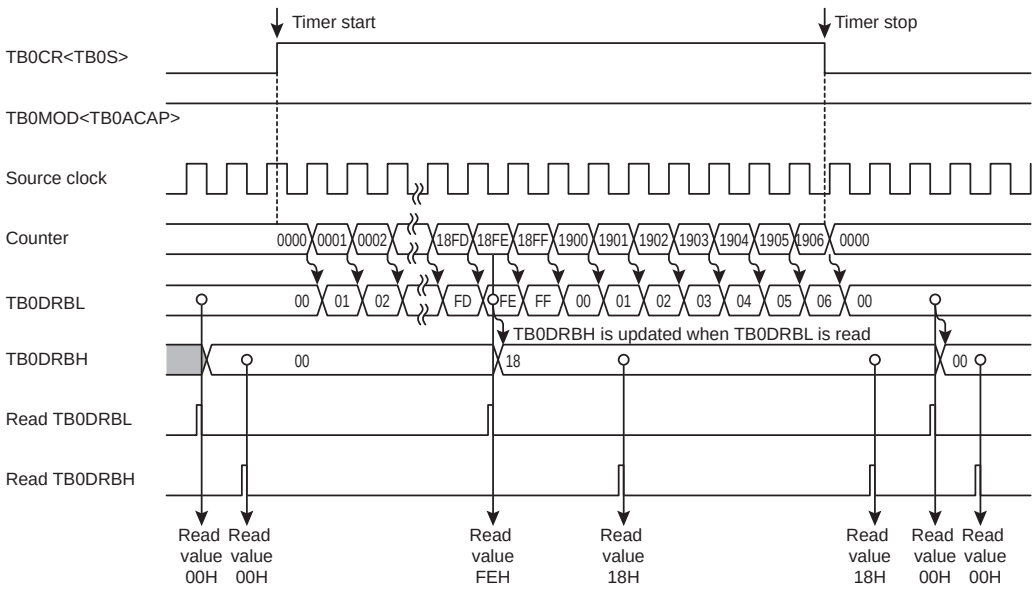


Figure 14-3 Timer Mode Timing Chart (Auto Capture)

Downloaded from Arrow.com.

14.4.2 External trigger timer mode

In the external trigger timer mode, the up counter starts counting when it is triggered by the input to the TCB0 pin.

14.4.2.1 Setting

Setting the operation mode selection TB0MOD<TB0M> to "100" activates the external trigger timer mode. Select the source clock at TB0MOD<TB0CK>.

Select the trigger edge at the trigger edge input selection TB0MOD<TB0TED>. Setting TB0MOD<TB0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge.

Note that this mode uses the TB0 input pin, and the TCB0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TB0CR<TB0S> to "1". After the timer is started, writing to TB0MOD and TB0CR<TB0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

14.4.2.2 Operation

After the timer is started, when the selected trigger edge is input to the TCB0 pin, the up counter increments according to the selected source clock. When a match between the up counter value and the value set to timer register A (TB0DRA) is detected, an INTTCB0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting.

When TB0MOD<TB0METT> is "1" and the edge opposite to the selected trigger edge is detected, the up counter stops counting and is cleared to "0x0000". Subsequently, when the selected trigger edge is detected, the up counter restarts counting. In this mode, an interrupt request can be generated by detecting that the input pulse exceeds a certain pulse width. If TB0MOD<TB0METT> is "0", the detection of the selected edge and the opposite edge is ignored during the period from the detection of the specified trigger edge and the start of counting through until the match detection.

Setting TB0CR<TB0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

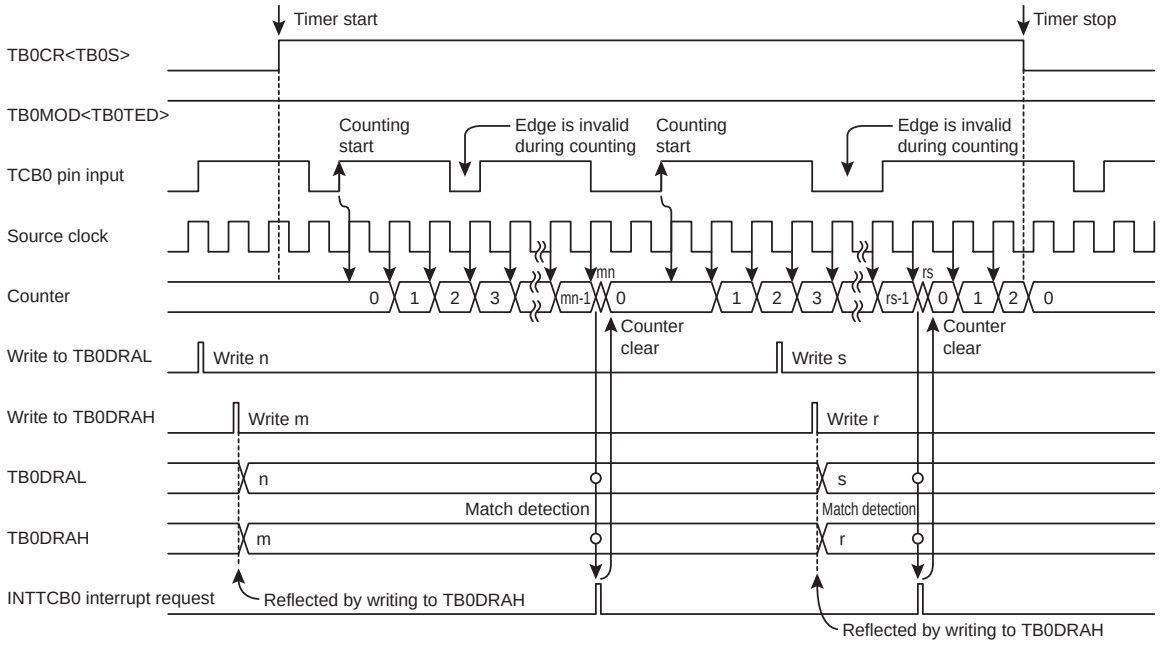
14.4.2.3 Auto capture

Refer to "14.4.1.3 Auto capture".

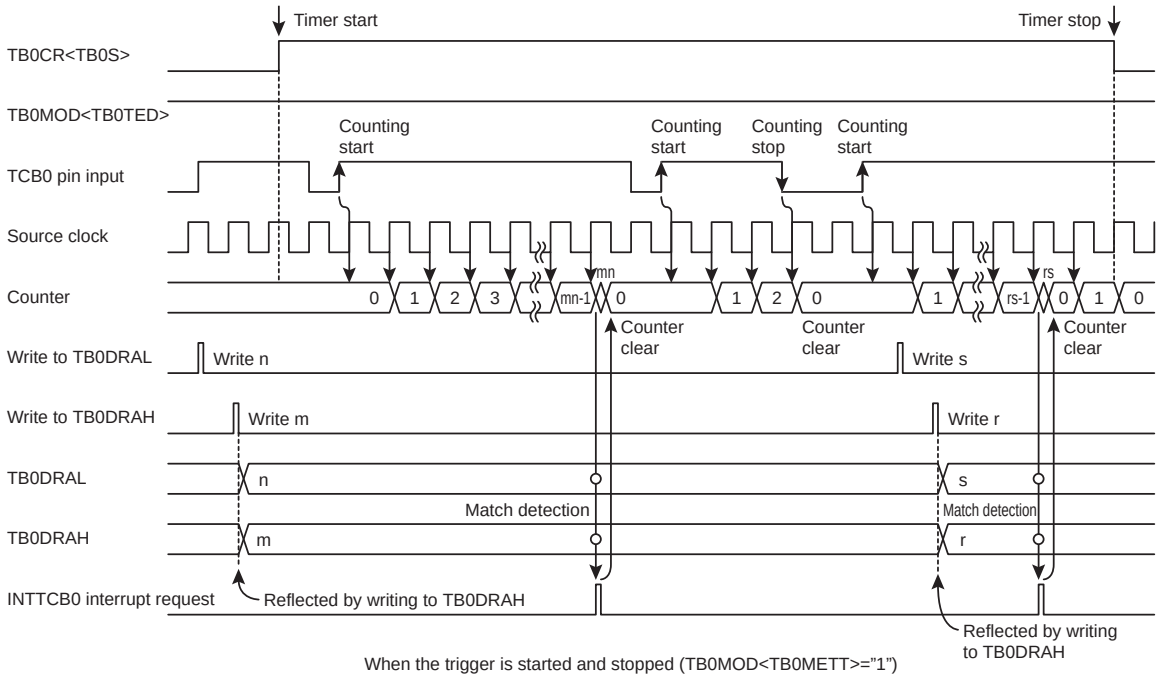
14.4.2.4 Register buffer configuration

Refer to "14.4.1.4 Register buffer configuration".

Downloaded from Arrow.com.



When the trigger is started (TB0MOD<TB0METT>="0")



When the trigger is started and stopped (TB0MOD<TB0METT>="1")

Figure 14-4 External Trigger Timer Timing Chart

Downloaded from Arrow.com.

14.4.3 Event counter mode

In the event counter mode, the up counter counts up at the edge of the input to the TCB0 pin.

14.4.3.1 Setting

Setting the operation mode selection TB0MOD<TB0M> to "010" activates the event counter mode.

Set the trigger edge at the external trigger input selection TB0MOD<TB0TED>. Setting TB0MOD<TB0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge for counting up.

Note that this mode uses the TB0 input pin, and the TCB0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TB0CR<TB0S> to "1". After the timer is started, writing to TB0MOD and TB0CR<TB0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

14.4.3.2 Operation

After the event counter mode is started, when the selected trigger edge is input to the TCB0 pin, the up counter increments.

When a match between the up counter value and the value set to timer register A (TB0DRA) is detected, an INTTCB0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter continues counting and counts up at each edge of the input to the TCB0 pin. Setting TB0CR<TB0S> to "0" during the operation causes the up counter to stop counting and be cleared to "0x0000".

The maximum frequency to be supplied is fcgck/2 [Hz] (in the NORMAL 1/2 or IDLE 1/2 mode) or fs/2 [Hz] (in the SLOW 1/2 or SLEEP 1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

14.4.3.3 Auto capture

Refer to "14.4.1.3 Auto capture".

14.4.3.4 Register buffer configuration

Refer to "14.4.1.4 Register buffer configuration".

Downloaded from Arrow.com.

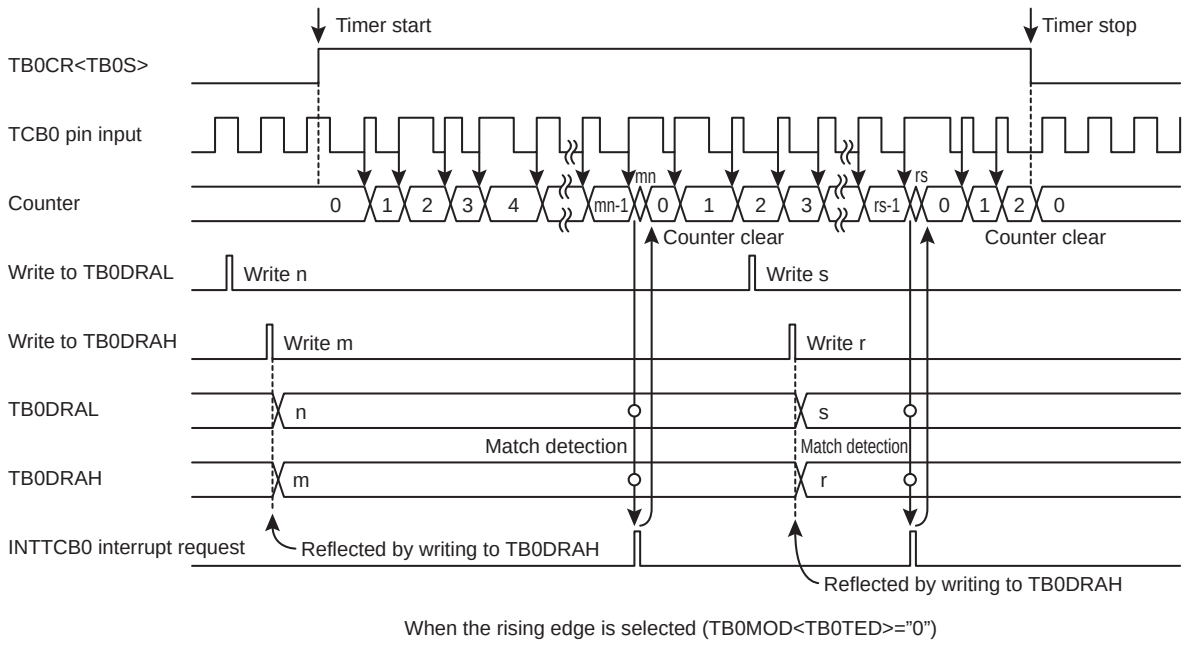


Figure 14-5 Event Count Mode Timing Chart

Downloaded from Arrow.com.

14.4.4 Window mode

In the window mode, the up counter counts up at the rising edge of the pulse that is logical anded product of the input pulse to the TCB0 pin (window pulse) and the internal clock.

14.4.4.1 Setting

Setting the operation mode selection TB0MOD<TB0M> to "101" activates the window mode. Select the source clock at TB0MOD<TB0CK>.

Select the window pulse level at the trigger edge input selection TB0MOD<TB0TED>. Setting TB0MOD<TB0TED> to "0" enables counting up as long as the window pulse is at the "H" level. Setting TB0MOD<TB0TED> to "1" enables counting up as long as the window pulse is at the "L" level.

Note that this mode uses the TB0 input pin, and the TCB0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TB0CR<TB0S> to "1". After the timer is started, writing to TB0MOD and TB0CR<TB0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

14.4.4.2 Operation

After the operation is started, when the level selected at TB0MOD<TB0TED> is input to the TCB0 pin, the up counter increments according to the source clock selected at TB0MOD<TB0CK>. When a match between the up counter value and the value set to timer register A (TB0DRA) is detected, an INTTCB0 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting.

The maximum frequency to be supplied must be slow enough for the program to analyze the count value. Define a frequency pulse that is sufficiently lower than the programmed internal source clock.

Setting TB0CR<TB0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

14.4.4.3 Auto capture

Refer to "14.4.1.3 Auto capture".

14.4.4.4 Register buffer configuration

Refer to "14.4.1.4 Register buffer configuration".

Downloaded from Arrow.com.

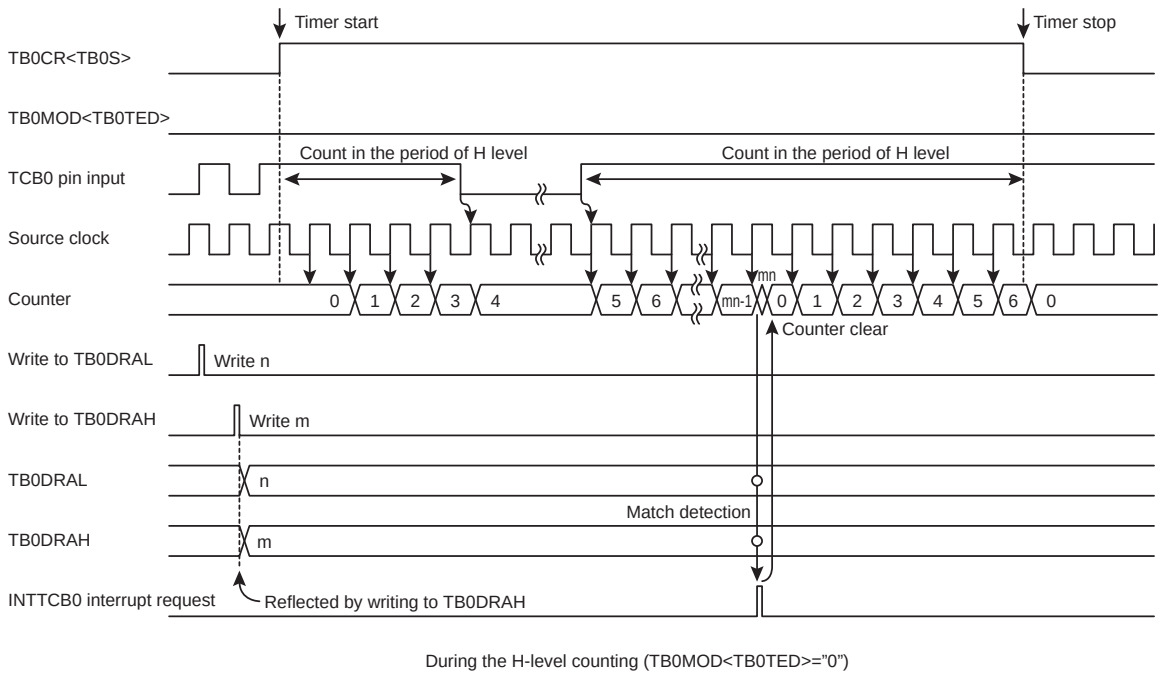


Figure 14-6 Window Mode Timing Chart

Downloaded from Arrow.com.

14.4.5 Pulse width measurement mode

In the pulse width measurement mode, the up counter starts counting at the rising/falling edge(s) of the input to the TCB0 pin and measures the input pulse width based on the internal clock.

14.4.5.1 Setting

Setting the operation mode selection TB0MOD<TB0M> to "110" activates the pulse width measurement mode. Select the source clock at TB0MOD<TB0CK>.

Select the trigger edge at the trigger edge input selection TB0MOD<TB0TED>. Setting TB0MOD<TB0TED> to "0" selects the rising edge, and setting it to "1" selects the falling edge as a trigger to start the capture.

The operation after capturing is determined by the pulse width measurement mode control TB0MOD<TB0MCAP>. Setting TB0MOD<TB0MCAP> to "0" selects the double-edge capture. Setting TB0MOD<TB0MCAP> to "1" selects the single-edge capture.

The operation to be executed in case of an overflow of the up counter can be selected at the overflow interrupt control TB0CR<TB0OVE>. Setting TB0OVE to "1" makes an INTTCB0 interrupt request occur in case of an overflow. Setting TB0OVE to "0" makes no INTTCB0 interrupt request occur in case of an overflow.

Note that this mode uses the TB0 input pin, and the TCB0 pin must be set to the input mode beforehand in port settings.

The operation is started by setting TB0CR<TB0S> to "1". In this time, TB0DRA and TB0DRB register are initialized to "0x0000". After the timer is started, writing to TB0MOD and TB0CR<TB0OVE> is disabled. Be sure to complete the required mode settings before starting the timer.

14.4.5.2 Operation

After the timer is started, when the selected trigger edge (start edge) is input to the TCB0 pin, INTTCB0 interrupt request is generated, and then the up counter increments according to the selected source clock. Subsequently, when the edge opposite to the selected edge is detected, the up counter value is captured into TB0DRB, an INTTCB0 interrupt request is generated, and TB0SR<TB0CPFB> is set to "1". Depending on the TB0MOD<TB0MCAP> setting, the operation differs as follows:

- Double-edge capture (When TB0MOD<TB0MCAP> is "0")

  The up counter continues counting up after the edge opposite to the selected edge is detected. Subsequently, when the selected trigger edge is input, the up counter value is captured into TB0DRA, an INTTCB0 interrupt request is generated, and TB0SR<TB0CPFA> is set to "1". At this time, the up counter is cleared to "0x0000".

- Single-edge capture (When TB0MOD<TB0MCAP> is "1")

  The up counter stops counting up and is cleared to "0x0000" when the edge opposite to the selected edge is detected. Subsequently, when the start edge is input, INTTCB0 interrupt request is generated, and then the up counter restarts increment.

When the up counter overflows during capturing, the overflow flag TB0SR<TB0OVF> is set to "1". At this time, an INTTCB0 interrupt request occurs if the overflow interrupt control TB0CR<TB0OVE> is set to "1".

The capture completion flags (TB0SR<TB0CPFA, TB0CPFB> and the overflow flag (TB0SR<TB0OVF>) are cleared to "0" automatically when TB0SR is read.

Downloaded from Arrow.com.

The captured value must be read from TB0DRB (and also from TB0DRA for the double-edge capture) before the next trigger edge is detected. If the captured value is not read, it becomes undefined. TB0DRA and TB0DRB must be read by using a 16-bit access instruction.

Setting TB0CR<TB0S> to "0" during the timer operation causes the up counter to stop counting and be cleared to "0x0000".

Note 1: After the timer is started, if the edge opposite to the selected trigger edge is detected first, no capture is executed and no INTTCB0 interrupt request occurs. In this case, the capture starts when the selected trigger edge is detected next.

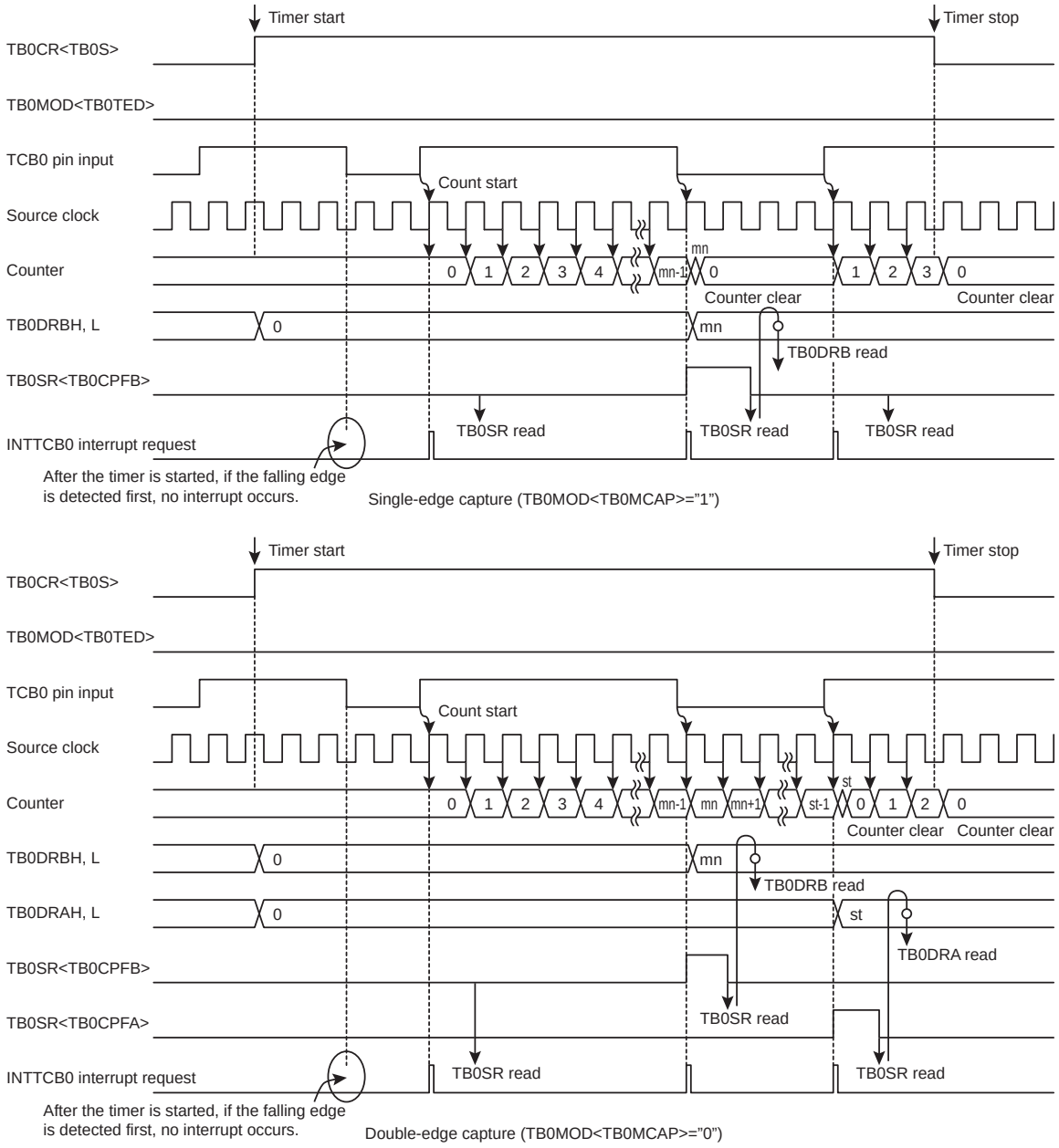
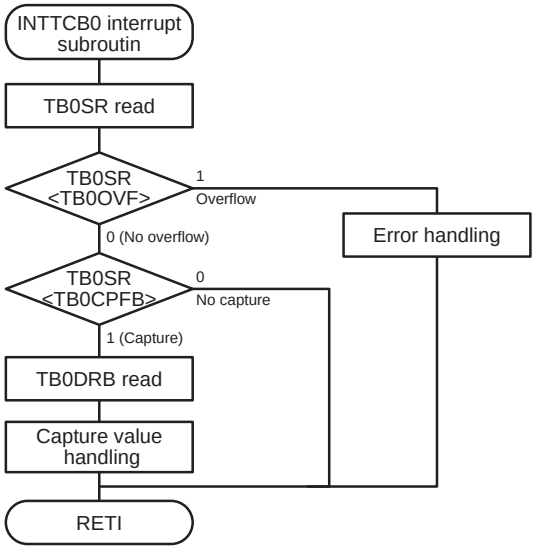


Figure 14-7 Pulse Width Measurement Mode Timing Chart

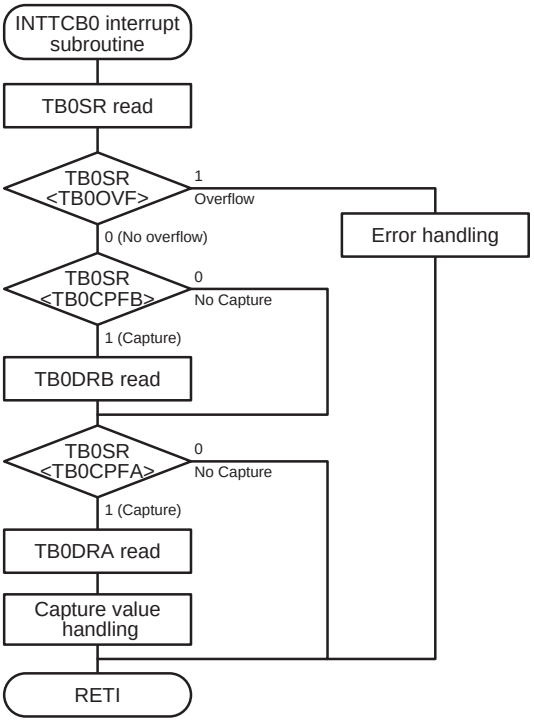
14.4.5.3 Capture process

Figure 14-8 shows an example of the capture process for INTTCB0 interrupt subroutine. The capture edge or overflow state can be easily judged by status register (TB0SR).

Downloaded from Arrow.com.



Interrupt process for single-edge capture



Interrupt process for double-edge capture

Figure 14-8 Example of capture process

Downloaded from Arrow.com.

14.4.6 Programmable pulse generate (PPG) mode

In the PPG output mode, an arbitrary duty pulse is output by two timer registers.

14.4.6.1 Setting

Setting the operation mode selection TB0MOD<TB0M> to "011" or "111" activates the PPG output mode. Setting TB0MOD<TB0M> to "011" selects the software start, and setting it to "111" selects the external trigger start (The external trigger start can be used only when TCA2 is used). Select the source clock at TB0MOD<TB0CK>. Select continuous or one-shot PPG output at TB0CR<TB0MPPG>.

Set the PPG output cycle at TB0DRA and set the time until the output is reversed first at TB0DRB. Be sure to set register values so that TB0DRA is larger than TB0DRB.

If you use the external trigger start, select the trigger edge at the trigger edge input selection TB0MOD<TB0TED>. Setting TB0MOD<TB0TED> to "0" selects the rising edge. Setting TB0MOD<TB0TED> to "1" selects the falling edge.

Note that this mode uses the TCB0 pin and the $\overline{\text{PPGB0}}$ pin. The TCB0 pin must be set to the input mode and the $\overline{\text{PPGB0}}$ pin must be set to the output mode beforehand in port settings.

Set the initial state of the $\overline{\text{PPGB0}}$ pin at the timer flip-flop TB0CR<TB0TFF>. Setting TB0CR<TB0TFF> to "1" selects the "H" level as the initial state of the $\overline{\text{PPGB0}}$ pin. Setting TB0CR<TB0TFF> to "0" selects the "L" level as the initial state of the $\overline{\text{PPGB0}}$ pin.

The operation is started by setting TB0CR<TB0S> to "1". After the timer is started, writing to TB0MOD and TB0CR<TB0OVE, TB0TFF> is disabled. Be sure to complete the required mode settings before starting the timer.

14.4.6.2 Operation

In the external trigger start mode, after the timer is started, when the selected trigger edge is input to the TCB0 pin, the up counter increments according to the selected source clock. In the software start mode, Aafter the timer is started, the up counter increments immediately without waiting for the edge.

When a match between the up counter value and the value set to timer register B (TB0DRB) is detected, the $\overline{\text{PPGB0}}$ pin is changed to the "H" level if TB0CR<TB0TFF> is "0", or the $\overline{\text{PPGB0}}$ pin is changed to the "L" level if TB0CR<TB0TFF> is "1".

Subsequently, the up counter continues counting. When a match between the up counter value and the value set to timer register A (TB0DRA) is detected, the $\overline{\text{PPGB0}}$ pin is changed to the "L" level if TB0CR<TB0TEFF> is "0", or the $\overline{\text{PPGB0}}$ pin is changed to the "H" level if TB0CR<TB0TFF> is "1". At this time, an INTTCB0 interrupt request occurs. If the PPG output control TB0CR<TB0MPPG> is set to "1" (one-shot), TB0CR<TB0S> is automatically cleared to "0" and the timer stops.

If TB0CR<TB0MPPG> is set to "0" (continuous), the up counter is cleared to "0x0000" and continues counting and PPG output. When TB0CR<TB0S> is set to "0" (including the auto stop by the one-shot operation) during the PPG output, the $\overline{\text{PPGB0}}$ pin returns to the level set in TB0CR<TB0TFF>.

TB0CR<TB0MPPG> can be changed during the operation. Changing TB0CR<TB0MPPG> from "1" to "0" during the operation cancels the one-shot operation and enables the continuous operation. Changing TB0CR<TB0MPPG> from "0" to "1" during the operation clears TB0CR<TB0S> to "0" and stops the timer automatically after the current pulse output is completed.

Timer registers A and B can be set to the double buffer. Setting TB0CR<TB0DBF> to "1" enables the double buffer. When the values set to TB0DRA and TB0DRB are changed during the PPG output with the double buffer enabled, the writing to TB0DRA and TB0DRB will not immediately become effective but will become effective when a match between TB0DRA and the up counter is detected. If the double buffer is dis-

Downloaded from Arrow.com.

abled, the writing to TB0DRA and TB0DRB will become effective immediately. If the written value is smaller than the up counter value, the up counter overflows. After a cycle, the counter match process is executed to reverse the output.

14.4.6.3 Register buffer configuration

(1) Temporary buffer

The TMP89FW24A contains an 8-bit temporary buffer. When a write instruction is executed on TB0DRAL (TB0DRBL), the data is first stored into this temporary buffer, whether the double buffer is enabled or disabled. Subsequently, when a write instruction is executed on TB0DRAH (TB0DRBH), the set value is stored into the double buffer or TB0DRAH (TB0DRBH). At the same time, the set value in the temporary buffer is stored into the double buffer or TB0DRAL (TB0DRBL). (This structure is designed to enable the set values of the lower-level register and the higher-level register simultaneously.) Therefore, when setting data to TB0DRA (TB0DRB), be sure to write the data into TB0DRAL and TB0DRAH (TB0DRBL and TB0DRBH) in this order.

See Figure 14-1 for the temporary buffer configuration.

(2) Double buffer

In the TMP89FW24A, the double buffer can be used by setting TB0CR<TB0DBF>. Setting TB0CR<TB0DBF> to "0" disables the double buffer. Setting TB0CR<TB0DBF> to "1" enables the double buffer.

See Figure 14-1 for the double buffer configuration.

- When the double buffer is enabled

  When a write instruction is executed on TB0DRAH (TB0DRBH) during the timer operation, the set value is first stored into the double buffer, and TB0DRAH/L are not updated immediately. TB0DRAH/L (TB0DRBH/L) compare the last set values to the counter value. If a match is detected, an INTTCB0 interrupt request is generated and the double buffer set value is stored into TB0DRAH/L (TB0DRBH/L). Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on TB0DRAH/L (TB0DRBH/L), the double buffer value (the last set value) is read, not the TB0DRAH/L (TB0DRBH/L) values (the current effective values).

  When a write instruction is executed on TB0DRAH/L (TB0DRBH/L) while the timer is stopped, the set value is immediately stored into both the double buffer and TB0DRAH/L (TB0DRBH/L).

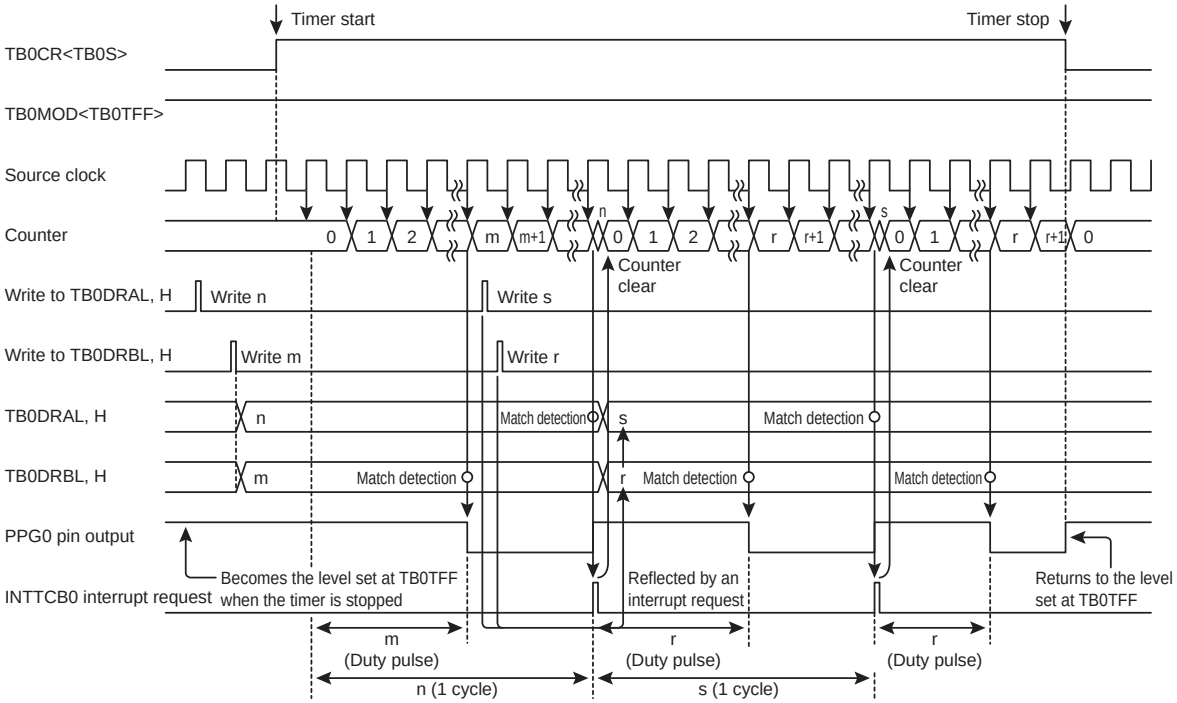
- When the double buffer is disabled

  When a write instruction is executed on TB0DRAH (TB0DRBH) during the timer operation, the set value is immediately stored in TB0DRAH/L (TB0DRBH/L). Subsequently, the match detection is executed using a new set value.

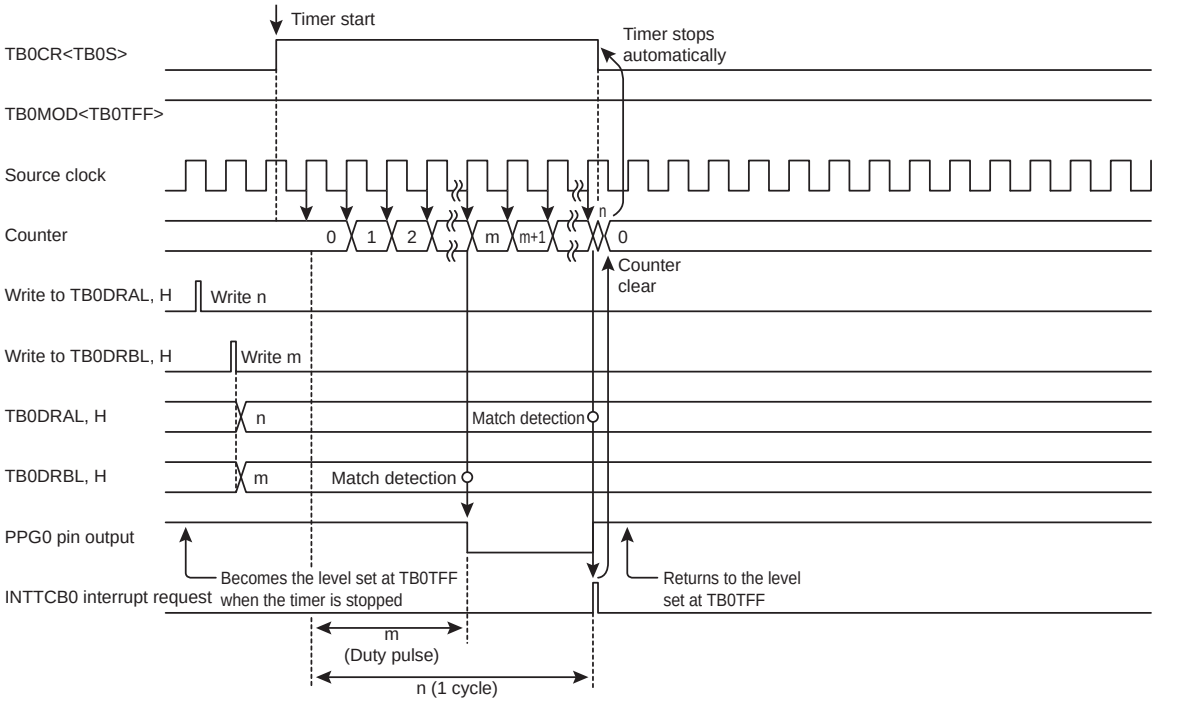
  If the values set to TB0DRAH/L (TB0DRBH/L) are smaller than the up counter value, the up counter overflows and the match detection is executed using a new set value. As a result, the output pulse width may be longer than the set time. If that is a problem, enable the double buffer.

  When a write instruction is executed on TB0DRAH/L (TB0DRBH/L) while the timer is stopped, the set value is immediately stored into TB0DRAH/L (TB0DRBH/L).

Downloaded from Arrow.com.



Continuous (TB0CR<TB0MPPG>="0")
Double buffer (TB0MOD<TB0DBE>="1")



One-shot (TB0CR<TB0MPPG>="1")

Figure 14-9 PPG Mode Timing Chart

Downloaded from Arrow.com.

14.5 Noise Canceller

The digital noise canceller can be used in the operation modes that use the TCB0 pin.

14.5.1 Setting

When the digital noise canceller is used, the input level is sampled at the sampling intervals set at TB0CR<TB0NC>. When the same level is detected three times consecutively, the level of the input to the timer is changed.

Setting TB0CR<TB0NC> to any values than "00" allows the noise canceller to start operation, regardless of the TB0CR<TB0S> value.

When the noise canceller is used, allow the timer to start after a period of time that is equal to four times the sampling interval after TB0CR<TB0NC> is set has elapsed. This stabilizes the input signal.

Set TB0CR<TB0NC> while the timer is stopped (TB0CR<TB0S> = "0"). When TB0CR<TB0S> is "1", writing is ignored.

In the SLOW 1/2 or SLEEP 1 mode, setting TB0CR<TB0NC> to "11" selects fs/2 as the source clock for the operation. Setting TB0CR<TB0NC> to "00" disables the noise canceller. Setting TB0CR<TB0NC> to "01" or "10" disables the TCB0 pin input.

Table 14-4 Noise Cancel Time (fcgck = 10 [MHz])

TB0NC	Sampling interval	Time removed as noise	Time regarded as signal
00	None	-	-
01	200 ns (2/fcgck)	600 ns or less	800 ns or more
10	400 ns (4/fcgck)	1.2 μs or less	1.6 μs or more
11	25.6 μs (256/fcgck)	76.8 μs or less	102.4 μs or more

Downloaded from Arrow.com.

15. 10-Bit Timer/Counter (TCC)

The TMP89FW24A contains 1 channels of high-performance 10-bit timer counters (TCC).

The 10-BitTimer/CounterC0 has one trigger input (TCC0) for start/stop/clear/capture of the counter.Timer has two PPG outputs (PPGC01,PPGC02) that can perform synchronous operation or individual operation.Timer has one emergency input(EMG0) for stop PPG output.

Table 15-1 SFR Address Assignment (The1)

	TCxCR1 (Address)	TCxCR2 (Address)	TCxCR3 (Address)
Timer/Counter C0	TC0CR1 (0x00E98)	TC0CR2 (0x00E99)	TC0CR3 (0x00E9A)

Table 15-2 SFR Address Assignment(The2)

	TCxDRA (Address)	TCxDRB (Address)	TCxDRC (Address)	TCxDRD (Address)	TCxDRE (Address)	TCxCAPA (Address)	TCxCAPB (Address)	Low power consumption register (Address)
Timer/Counter C0	TC0DRA (0x00E9C) (0x00E9B)	TC0DRB (0x00E9E) (0x00E9D)	TC0DRC (0x00EA0) (0x00E9F)	TC0DRD (0x00EA2) (0x00EA1)	TC0DRE (0x00EA4) (0x00EA3)	TC0CAPA (0x00EA6) (0x00EA5)	TC0CAPB (0x00EA8) (0x00EA7)	POFFCR0 <TCC0EN> (0x00F74)

Table 15-3 Pin Names

	Timer input pin	PPG output pin	PPG output pin	EMG input pin
Timer/CounterC0	TCC0 pin	$\overline{\text{PPGC01}}$ pin	$\overline{\text{PPGC02}}$ pin	$\overline{\text{EMG0}}$ pin

Downloaded from Arrow.com.

15.1 Configuration

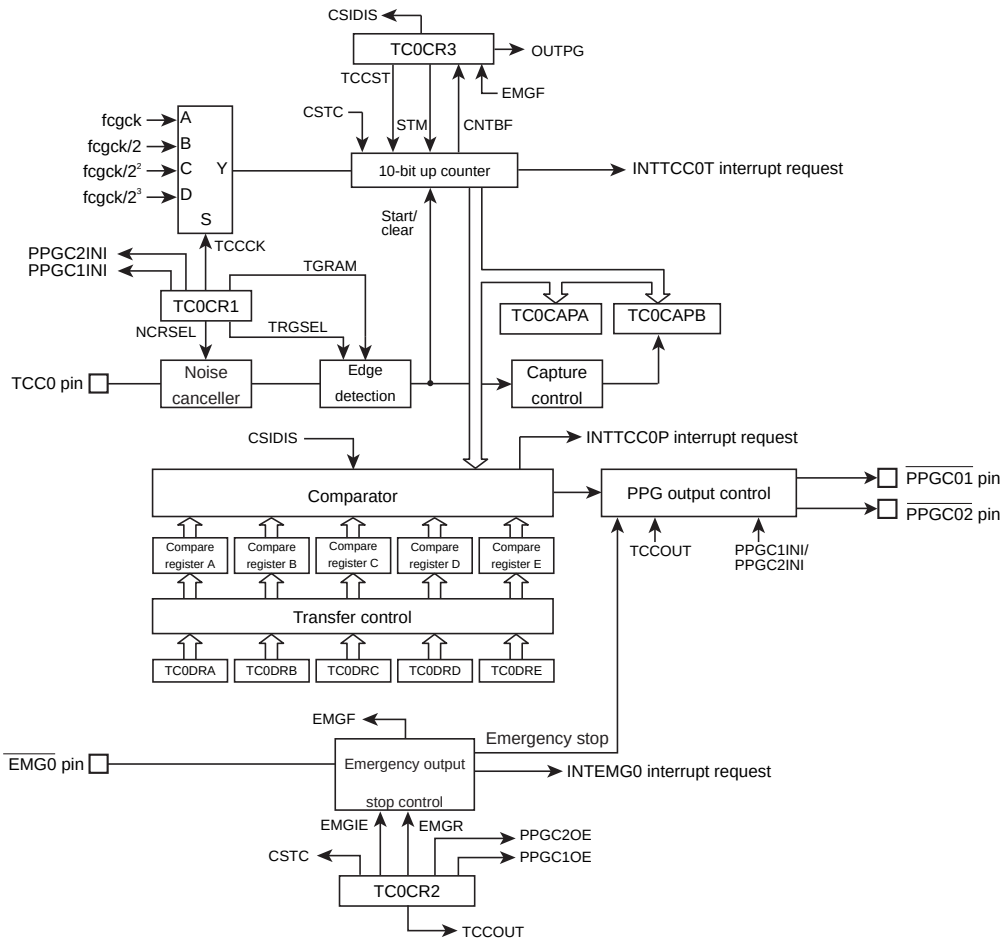


Figure 15-1 10-Bit Timer/Counter C0

Downloaded from Arrow.com.

15.2 Control

Timer/counter C0 is controlled by low power consumption registers (POFFCR0),timer/counter control register 1 (TC0CR1), timer/counter control register 2(TC0CR2), timer/counter control register 3 (TC0CR3), 10-bit dead time 1 setup register (TC0DRA), pulse width 1 setup register (TC0DRB), period setup register (TC0DRC), dead time 2 setup register (TC0DRD), pulse width 2 setup register (TC0DRE), and two capture value registers (TC0CA-PA and TC0CAPB).

Low power consumption register 0

POFFCR0		7	6	5	4	3	2	1	0
(0x00F74)	Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCB0EN	TCA0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 Control	0 1	Disable Enable
TC001EN	TC00,01 Control	0 1	Disable Enable
TCC0EN	TCC0 Control	0 1	Disable Enable
TCB0EN	TCB0 Control	0 1	Disable Enable
TCA0EN	TCA0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

Timer/Counter C0 Control Register 1

TC0CR1 (0x00E98)	7	6	5	4	3	2	1	0
Bit Symbol	TRGAM	TRGSEL	PPGC2INI	PPGC1INI	NCRSEL		TCCCK	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TRGAM	Trigger edge acceptance control	0: 1:	Always accept trigger edges. (Refer to the setting of TC0CR2<PPGC2OE, PPGC1OE>)
TRGSEL	Select a trigger start edge.	0: 1:	Start on trigger falling edge. Start on trigger rising edge.
PPGC2INI	Specify the initial value of PPG2 output.	0: 1:	Low (Positive logic) High (Negative logic)
PPGC1INI	Specify the initial value of PPG1 output.	0: 1:	Low (Positive logic) High (Negative logic)
NCRSEL	Select the duration of noise elimination for TCC0 input (after passing through the flip-flop).	00: 01: 10: 11:	Eliminate pulses shorter than 16/fcgck [s] as noise. Eliminate pulses shorter than 8/fcgck [s] as noise. Eliminate pulses shorter than 4/fcgck [s] as noise. Do not eliminate noise. (Note)
TCCCK	Select a source clock	00: 01: 10: 11:	fcgck [Hz] fcgck/2 [Hz] fcgck/2^2 [Hz] fcgck/2^3 [Hz]

Note: Due to the circuit configuration, a pulse shorter than 1/fcgck may be eliminated as noise or accepted as a trigger.

Timer/Counter C0 Control Register 2

TC0CR2 (0x00E99)	7	6	5	4	3	2	1	0
Bit Symbol	EMGR	EMGIE	PPGC2OE	PPGC1OE	CSTC		TCCOUT	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

EMGR	Cancel the emergency output stop state.	0: 1:	- Cancel the emergency output stop state. (Upon canceling the state,this bit is automatically cleared to 0.)	
EMGIE	Enable/disable input on the $\overline{EMG0}$ pin.	0: 1:	Disable input. Enable input.	
PPGC2OE	Trigger edge accept control during the PPGC02 output.		When TC0CR1<TRGAM> is "0"	When TC0CR1<TRGAM> is "1"
		0: 1:	Always accept trigger edges. Always accept trigger edges.	Always accept trigger edges. Do not accept trigger edges during active period.
PPGC1OE	Trigger edge accept control during the PPGC01 output.	0: 1:	Always accept trigger edges. Always accept trigger edges.	Always accept trigger edges. Do not accept trigger edges during active period.
CSTC	Select a count start mode.	00: 01: 10: 11:	Command start and capture mode Command start and trigger start mode. Trigger start mode Reserved	
TCCOUT	Select an output waveform mode.	00: 01: 10: 11:	PPGC01/PPGC02 independent output Reserved Output with variable duty ratio Output with 50% duty ratio	

Downloaded from Arrow.com.

Timer/Counter C0 Control Register 3

TC0CR3		7	6	5	4	3	2	1	0
(0x00E9A	Bit Symbol	-	-	EMGF	CNTBF	CSIDIS	STM		TCCST
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

EMGF	Emergency output stop flag	0: 1:	Operating normally Output stopped in emergency	
CNTBF	Counting status flag	0: 1:	Counting stopped Counting in progress	
CSIDIS	Disable the first interrupt at upon a command start.	0:	Allow a periodic interrupt (INTTCC0P) to occur in the first period upon a command start.	
		1:	Do not allow a periodic interrupt (INTTCC0P) to occur in the first period upon a command start.	
STM	Select the state when stopped. Select continuous or one-time output.		TCCST = "0"	TCCST = "1"
		00:	Immediately stop and clear the counter with the output initialized.	Continuous output
		01:	Immediately stop and clear the counter with the output initialized.	Continuous output
		10:	Stop the counter after completing output in the current period.	One-time output
		11:	Reserved	
TCCST	Start/stop the timer.	0: 1:	Stop Start	

Note 1: The TC0CR1 and TC0CR2 registers should not be rewritten after a timer start (when TCCST is "1").

Note 2: Before attempting to modify the TC0CR1 or TC0CR2, clear TCCST and then check that CNTBF = 0 to determine that the timer is stopped.

Note 3: The TCCST bit only causes the timer to start or stop; it does not indicate the current operating state of the counter. Its value does not change automatically when counting starts or stops

Note 4: In command start and capture mode or command start and trigger start mode, writing 1 to TCCST causes the timer to restart immediately. It means that rewriting any bit other than TCCST in the TC0CR3 after a command start causes the rewriting of TCCST, resulting in the timer being restarted (PPG output is started from the initial state). When TCCST is set to 1, rewriting the TC0CR3 (Using a bit manipulation or LD instruction) clears the counter and restarts the timer.

Note 5: TC0CR2<EMGR> is always read as 0 even after 1 is written.

Note 6: Data registers are not updated by merely modifying the output mode with TC0CR2<TCCOUT>. After modifying the output mode, reconfigure data registers TC0DRA to TC0DRE. Ensure that the data registers are written in an appropriate order because they are not enabled until the upper byte of the TC0DRC is written.

Note 7: When a read instruction is executed on TC0CR3, bits 7 and 6 are read as "0".

Downloaded from Arrow.com.

Dead Time 1 Setup Register AH

TC0DRAH (0x00E9C)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRAH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Dead Time 1 Setup Register AL

TC0DRAL (0x00E9B)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRAL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 1 Setup Register BH

TC0DRBH (0x00E9E)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRAH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 1 Setup Register BL

TC0DRBL (0x00E9D)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRAL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Period Setup Register CH

TC0DRCH (0x00EA0)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRCH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Period Setup Register CL

TC0DRCL (0x00E9F)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRCL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Downloaded from Arrow.com.

Dead Time 2 Setup Register DH

TC0DRDH (0x00EA2)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DRDH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Dead Time 2 Setup Register DL

TC0DRDL (0x00EA1)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DRDL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 2 Setup Register EH

TC0DREH (0x00EA4)	15	14	13	12	11	10	9	8
Bit Symbol	TC0DREH							
Read/Write	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Pulse Width 2 Setup Register EL

TC0DREL (0x00EA3)	7	6	5	4	3	2	1	0
Bit Symbol	TC0DREL							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1: Data registers TC0DRA to TC0DRE have double-stage configuration, consisting of a data register that stores data written by an instruction and a compare register to be compared with the counter.

Note 2: When writing data to data registers TC0DRA to TC0DRE, first write the lower byte and then the upper byte.

Note 3: Unused bits (Bits 10 to 15) in the upper bytes of data registers TC0DRA to TC0DRE are not assigned specific register functions. These bits are always read as 0 even when a 1 is written.

Note 4: Values read from data registers TC0DRA to TC0DRE may differ from the actual PPG output waveforms due to their double-stage configuration.

Note 5: Data registers are not updated by merely modifying the output mode with TC0CR2<TCCOUT>. After modifying the output mode, reconfigure data registers TC0DRA to TC0DRE. Ensure that the data registers are written in an appropriate order because they are not enabled until the upper byte of the TC0DRC is written.

Downloaded from Arrow.com.

Rising-edge Capture Value Register AH

TC0CAPAH (0x00EA6)	15	14	13	12	11	10	9	8
Bit Symbol	TC0CAPAH							
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*

Rising-edge Capture Value Register AL

TC0CAPAL (0x00EA5)	7	6	5	4	3	2	1	0
Bit Symbol	TC0CAPAL							
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*

Rising-edge Capture Value Register BH

TC0CAPBH (0x00EA8)	15	14	13	12	11	10	9	8
Bit Symbol	TC0CAPBH							
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	*	*

Rising-edge Capture Value Register BL

TC0CAPBL (0x00EA7)	7	6	5	4	3	2	1	0
Bit Symbol	TC0CAPBL							
Read/Write	R	R	R	R	R	R	R	R
After reset	*	*	*	*	*	*	*	*

Note 1: Capture registers (TC0CAPA and TC0CAPB) must be read in the following order: Lower byte of the TC0CAPA, upper byte of the TC0CAPA, lower byte of the TC0CAPB, upper byte of the TC0CAPB.

Note 2: The next captured data is not updated by reading the TC0CAPA only. The TC0CAPB must also be read.

Note 3: It is possible to read the TC0CAPB only. Read the lower byte first.

Note 4: If a capture edge is not detected within a period, the previous capture value is maintained in the next period.

Note 5: If more than one capture edge is detected within a period, the capture value for the edge detected last is valid in the next period.

Note 6: When a read instruction is executed on TC0CAPA and TC0CAPB, bits 15 to 10 are read as "0".

Downloaded from Arrow.com.

15.3 Low Power Consumption Function

Timer counter C0 has the low power consumption register (POFFCR0) that saves power consumption when the timer is not used.

Setting POFFCR0<TCC0EN> to "0" disables the basic clock supply to timer counter C0 to save power. Note that this makes the timer unusable. Setting POFFCR0<TCC0EN> to "1" enables the basic clock supply to timer counter C0 and allows the timer to operate.

After reset, POFFCR0<TCC0EN> is initialized to "0", and this makes the timer unusable. When using the timer for the first time, be sure to set POFFCR0<TCC0EN> to "1" in the initial setting of the program (before the timer control register is operated).

Do not change POFFCR0<TCC0EN> to "0" during the timer operation. Otherwise timer counter C0 may operate unexpectedly.

Downloaded from Arrow.com.

15.4 Configuring Control and Data Registers

Configure control and data registers in the following order:

1. Configure mode settings: TC0CR1, TC0CR2
2. Configure data registers (Dead time, pulse width):TC0DRA, TC0DRB, TC0DRD, TC0DRE (only those required for selected mode)
3. Configure data registers (Period): TC0DRC
4. Configure timer start/stop:TC0CR3

- Data registers have double-stage configuration, consisting of a data register that stores data written by an instruction and a compare register to be compared with the counter.
- Data stored in a data register is processed according to the output mode specified in the TC0CR2<TCCOUT>,transferred to the compare register, and then used for comparison with the up counter.
- Data registers required for the specified output mode are used for data register processing and transfer to the compare register. Ensure that the output mode is specified in the TC0CR2<TCCOUT> before configuring data registers.
- Writing data to the upper byte of the TC0DRC causes a data transfer request to be issued for data in data registers TC0DRA to TC0DRE. If a counter match or clear occurs while that request is valid, the data is transferred to the compare register and becomes valid for comparison.
- If a data register is written more than once within a period, the data in the data register that was set when the upper byte of the TC0DRC was written is valid as data for the next period. The data in the data register written last in the first period will be valid for the period that follows the next period.

Downloaded from Arrow.com.

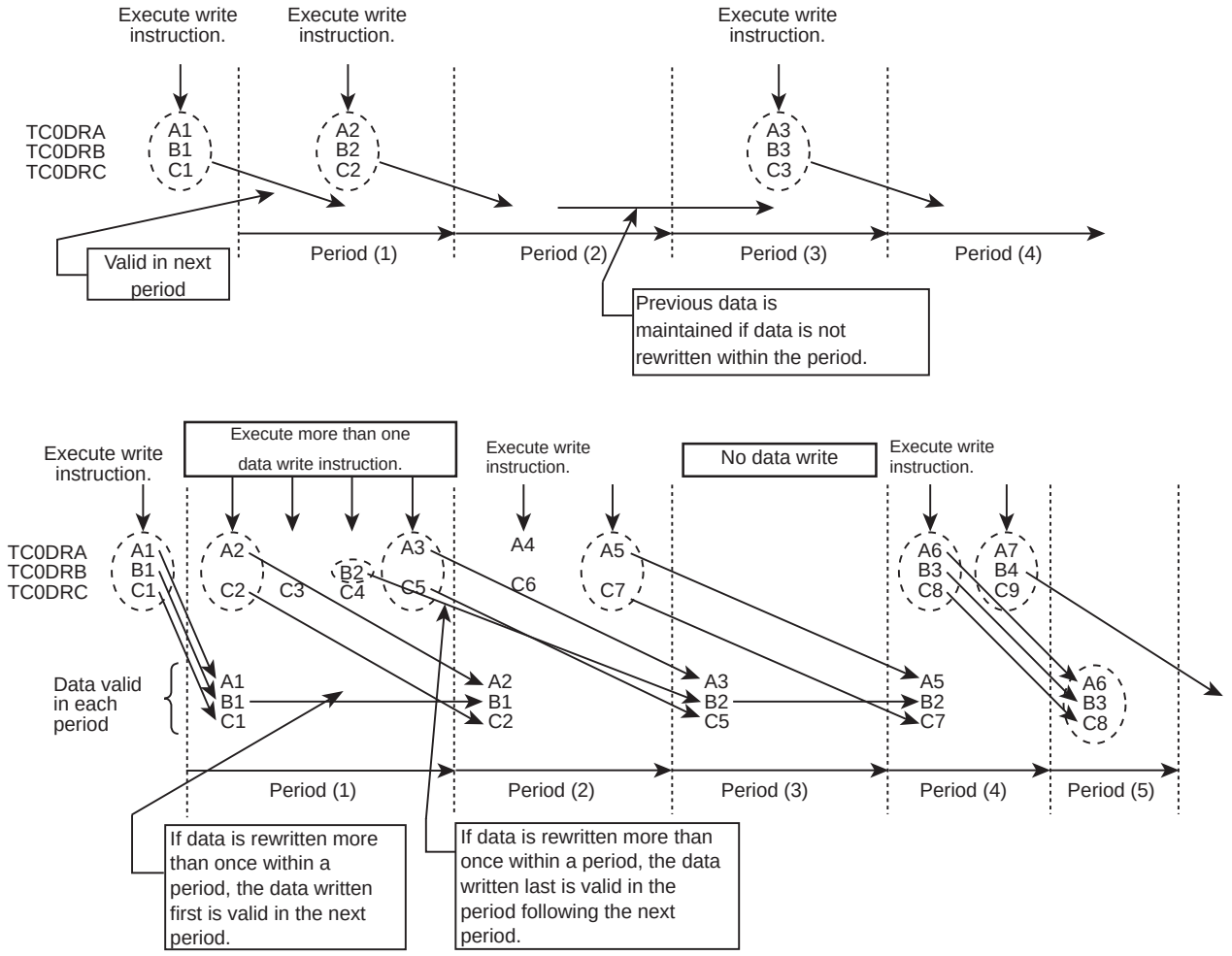


Figure 15-2 Example Configuration of Control/data Registers (1)

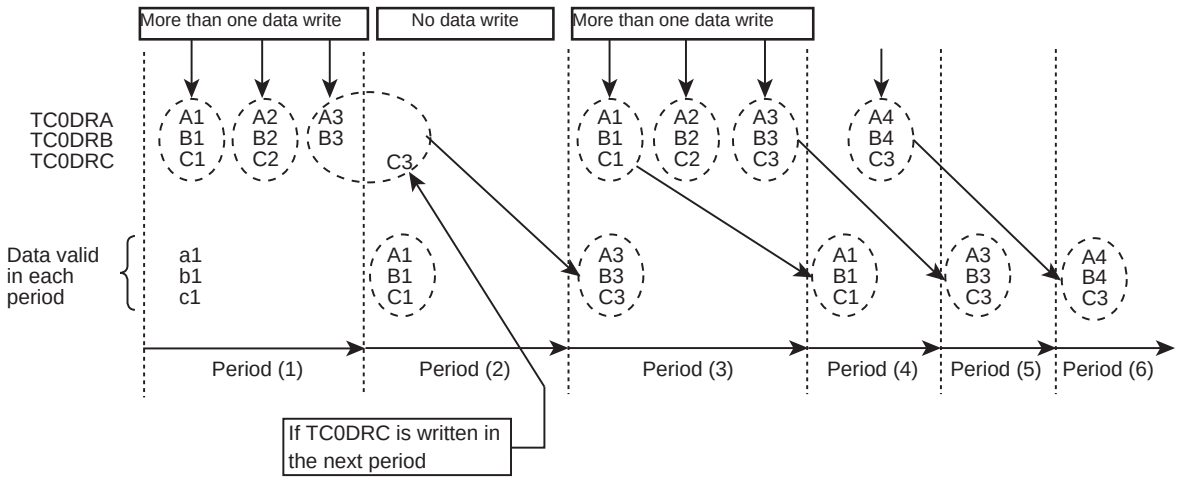


Figure 15-3 Example Configuration of Control/data Registers (2)

Downloaded from Arrow.com.

15.5 Features

15.5.1 Programmable pulse generator output (PPG output)

The $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ pins provide PPG outputs. The output waveform mode for PPG outputs is specified with TC0CR2<TCCOUT> and their waveforms are controlled by comparing the contents of the 10-bit up counter with the data set in data registers (TC0DRA to TC0DRE). Three output waveform modes are available:50% duty mode, variable duty mode, and $\overline{\text{PPGC01}}$/$\overline{\text{PPGC02}}$ independent mode.

15.5.1.1 50% duty mode

(1) Description

With a period specified in the TC0DRC, the $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ pins provide waveforms having a pulse width (Active duration) that equals a half the period.

The $\overline{\text{PPGC01}}$ output is active at the beginning of a period and becomes inactive at half the period. The $\overline{\text{PPGC02}}$ output is inactive at the beginning of a period, becomes active at half the period, and remains active until the end of the period.

If a dead time is specified in the TC0DRA, the pulse width (Active duration) is shortened by the dead time.

(2) Register settings

TC0CR2<TCCOUT> = “11”, TC0DRA = “dead time”, TC0DRC = “period”

(3) Valid range for data register values

- Period:

  0x002 ≤ TC0DRC ≤ 0x400

  (Writing 0x400 to TC0DRC results in 0x000 being read from it)

  When the value set in the TC0DRC is an odd number, the $\overline{\text{PPGC02}}$ pulse width is one count longer than the $\overline{\text{PPGC01}}$ pulse width.
- Dead time TC0DRA:

  0x000 ≤ TC0DRA < (TC0DRC ÷ 2)

  To specify no dead time, set the TC0DRA to 0x000.

Downloaded from Arrow.com.

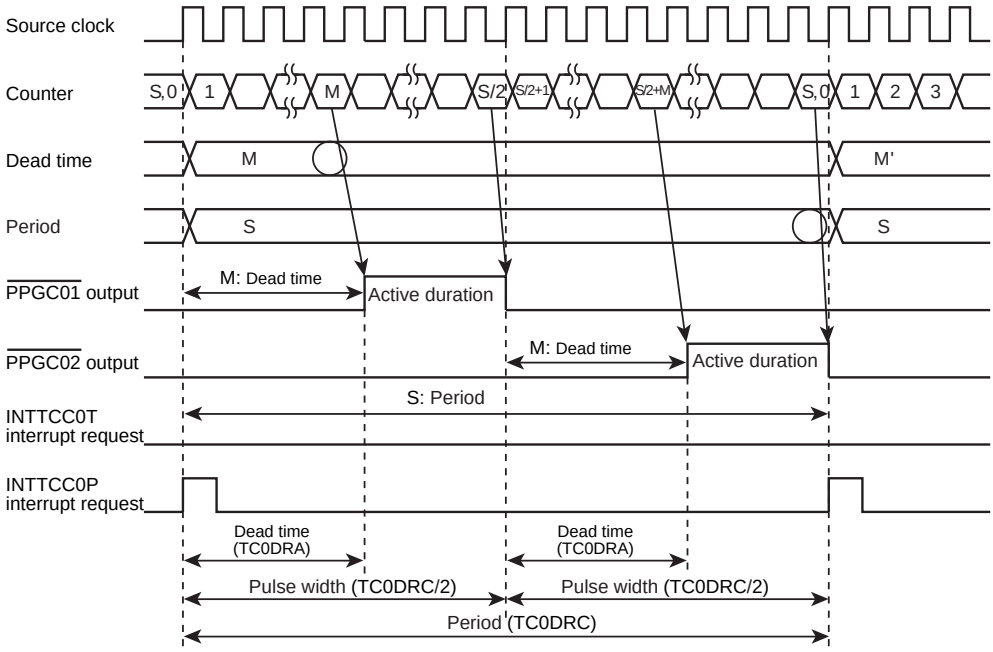


Figure 15-4 Example operation in 50% duty mode: Command and capture start, positive logic, continuous output

15.5.1.2 Variable duty mode

(1) Description

With a period specified in the TC0DRC and a pulse width in the TC0DRB, the $\overline{\text{PPGC01}}$ pin provides a waveform having the specified pulse width while the $\overline{\text{PPGC02}}$ pin provides a waveform having a pulse width that equals (TC0DRC . TC0DRB).

The $\overline{\text{PPGC01}}$ output is active at the beginning of a period, remains active during the pulse width specified in the TC0DRB, after which it is inactive until the end of the period. The $\overline{\text{PPGC02}}$ output is inactive at the beginning of a period, remains inactive during the pulse width specified in the TC0DRB,after which it is active until the end of the period, that is, during the pulse width of (TC0DRC .TC0DRB).

If a dead time is specified in the TC0DRA, the pulse width (Active duration) is shortened by the dead time.

(2) Register settings

TC0CR2<TCCOUT> = “10”

TC0DRA = “dead time”, TC0DRB = “pulse width”, TC0DRC = “period”

(3) Valid range for data register values

- Period:

  0x002 ≤ TC0DRB + TC0DRA < TC0DRC ≤ 0x400

  (Writing 0x400 to TC0DRC results in 0x000 being read from it.)
- Pulse width:

Downloaded from Arrow.com.

0x001 ≤ TC0DRB < TC0DRC

- Dead time:

  0x000 ≤ TC0DRA < TC0DRB

  0x000 ≤ TC0DRA < (TC0DRC − TC0DRB)

  (To specify no dead time, set the TC0DRA to 0x000.)

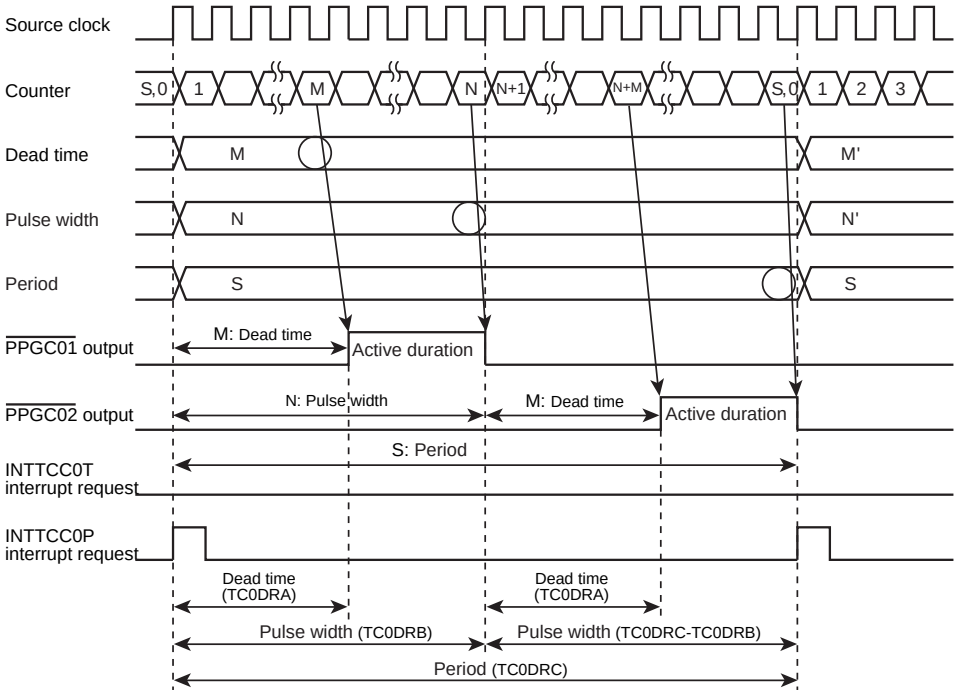


Figure 15-5 Example Operation in Variable Duty Mode: Command and Capture Start, Positive Logic, Continuous Output

15.5.1.3 PPGC01/PPGC02 independent mode

(1) Description

For the $\overline{\text{PPGC01}}$ output, specify the dead time in the TC0DRA and pulse width in the TC0DRB. For the $\overline{\text{PPGC02}}$ output, specify the dead time in the TC0DRD and pulse width in the TC0DRE. With a common period specified in the TC0DRC, the $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ pins provide waveforms having the specified pulse widths.

The $\overline{\text{PPGC01}}$ output is active at the beginning of a period, remains active during the pulse width specified in the TC0DRB, after which it is inactive until the end of the period.

The $\overline{\text{PPGC02}}$ output is active at the beginning of a period, remains active during the pulse width specified in the TC0DRE, after which it is inactive until the end of the period.

If a dead time is specified in the TC0DRA for the $\overline{\text{PPGC01}}$ output or in the TC0DRD for the $\overline{\text{PPGC02}}$ output, the pulse width (Active duration) is shortened by the dead time.

(2) Register settings

TC0CR2<TCCOUT> = “00”, TC0DRC = “period”,

TC0DRA = “PPGC01 dead time”, TC0DRB = “PPGC01 pulse width”,

Downloaded from Arrow.com.

TC0DRD = “PPGC02 dead time”, TC0DRE = “PPGC02 pulse width”

(3) Valid range for data register values

- Period:

  0x002 ≤ TC0DRC ≤ 0x400

  (Writing 0x400 to TC0DRC results in 0x000 being read from it.)
- Pulse width:

  0x001 ≤ TC0DRB ≤ 0x400

  (Writing 0x400 to TC0DRB results in 0x000 being read from it.)

  0x001 ≤ TC0DRE ≤ 0x400

  (Writing 0x400 to TC0DRE results in 0x000 being read from it.)
- Dead time:

  0x000 ≤ TC0DRA ≤ 0x3FF, where TC0DRA < TC0DRB ≤ TC0DRC

  0x000 ≤ TC0DRD ≤ 0x3FF, where TC0DRD < TC0DRE ≤ TC0DRC

  (To specify no dead time, write 0x000)

1. Settings for a duty ratio of 0%

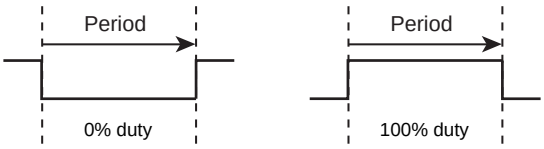
   0x002 ≤ TC0DRC ≤ TC0DRA ≤ 0x3FF ($\overline{\text{PPGC01}}$ output)

   0x002 ≤ TC0DRC ≤ TC0DRD ≤ 0x3FF ($\overline{\text{PPGC02}}$ output)

2. Settings for a duty ratio greater than 0%, up to 100%

   0x000 ≤ TC0DRA < TC0DRB ≤ TC0DRC ≤ 0x400 ($\overline{\text{PPGC01}}$ output)

   0x000 ≤ TC0DRD < TC0DRE ≤ TC0DRC ≤ 0x400 ($\overline{\text{PPGC02}}$ output)



Downloaded from Arrow.com.

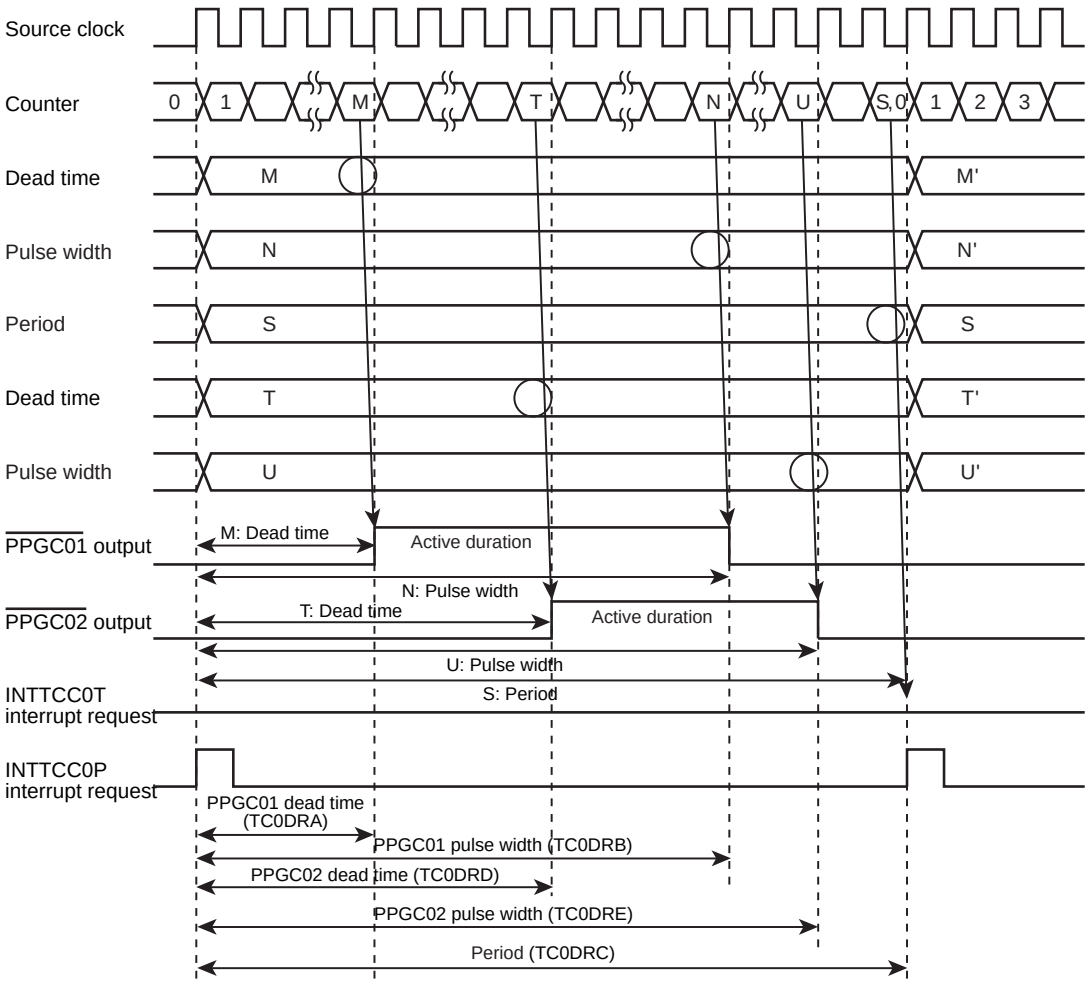


Figure 15-6 Example Operation in PPGC01/PPGC02 Independent Mode: Command and Capture Start, Positive Logic, Continuous Output

15.5.2 Starting a count

A count can be started by using a command or TCC0 pin input.

15.5.2.1 Command start and capture mode(TC0CR2<CSTC>="00")

(1) Description

Writing a 1 to TC0CR3<TCCST> causes the current count to be cleared and the counter to start counting. Once the count has reached a specified period, the counter is cleared. The counter subsequently restarts counting if TC0CR3<STM> specifies continuous mode; it stops counting if TC0CR3<STM> specifies one-time mode.

Writing a 1 to TC0CR3<TCCST> before the count reaches a period causes the counter to be cleared, after which it operates as specified with TC0CR3<STM>.

The count values at the rising and falling edges on the TCC0 pin can be stored in capture registers

(For more information of the capture, see 1.5.5.3 “Trigger capture” .)

Downloaded from Arrow.com.

(2) Register settings

- TC0CR2<CSTC> = “00” Command start and capture mode

- TC0CR3<STM> Continuous/one-time output

- TC0CR3<TCCST> = “1” Starts counting

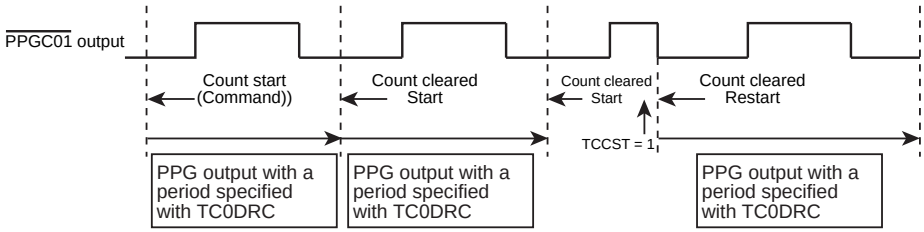


Figure 15-7 Example Operation in Command Start and Capture Mode

15.5.2.2 Command start and trigger start mode(TC0CR2<CSTC>="01")

(1) Description

Writing a 1 to TC0CR3<TCCST> causes the current count to be cleared and the counter to start counting. The operation is the same as that in command start and capture mode if there is no trigger input on the TCC0 pin. If an edge specified with the start edge selection field (TC0CR1<TRGSEL>) appears on the TCC0 pin,however, the timer starts counting. The counter is cleared and stopped while the TCC0 pin is driven to the specified clear/stop level. If the TCC0 pin is at the clear/stop level when a count start command is issued (1 is written to TC0CR3<TCCST>), counting does not start (INTTCC0P does not occur) until a trigger start edge appears, causing INTTCC0T to occur (A trigger input takes precedence over a command start).

Note: For more information on the acceptance of a trigger, see 15.5.4 “Trigger start/stop acceptance mode” .

(2) Register settings

- TC0CR2<CSTC> = “01” Command start and trigger start mode

- TC0CR1<TRGSEL> = Trigger selection

- TC0CR3<STM> Continuous/one-time output

- TC0CR3<TCCST> = “1” Starts counting

Downloaded from Arrow.com.

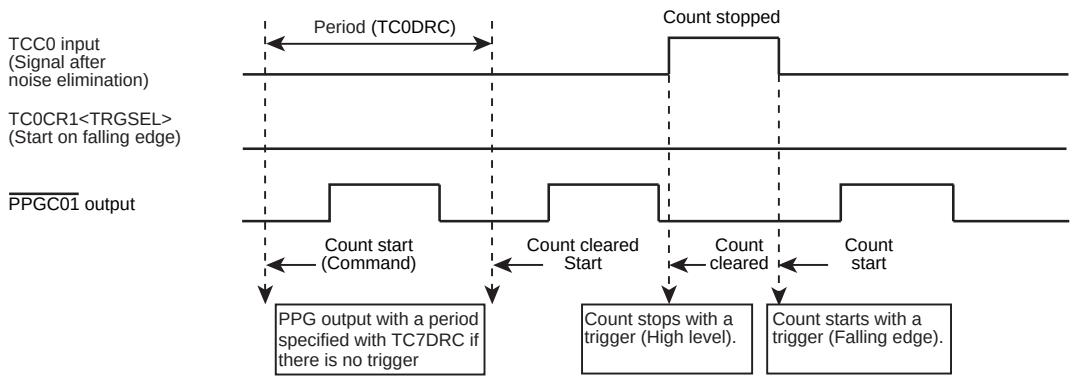


Figure 15-8 Example Operation in Command Start and Trigger Start Mode

15.5.2.3 Trigger start mode(TC0CR2<CSTC>="10")

(1) Description

If an edge specified with the start edge selection field (TC0CR1<TRGSEL>) appears on the TCC0 pin, the timer starts counting. The counter is cleared and stopped while the TCC0 pin is driven to the specified clear/stop level.

In trigger start mode, writing a 1 to TC0CR3<TCCST> is ignored and does not initialize the PPG output.

Note: For more information on the acceptance of a trigger, see 15.5.4 “Trigger start/stop acceptance mode”.

(2) Register settings

- TC0CR2<CSTC> = “10” Trigger start mode
- TC0CR1<TRGSEL> = Trigger selection
- TC0CR3<STM> Continuous/one-time output
- TC0CR3<TCCST> = “1” Starts waiting for a trigger on the TCC0 pin

Downloaded from Arrow.com.

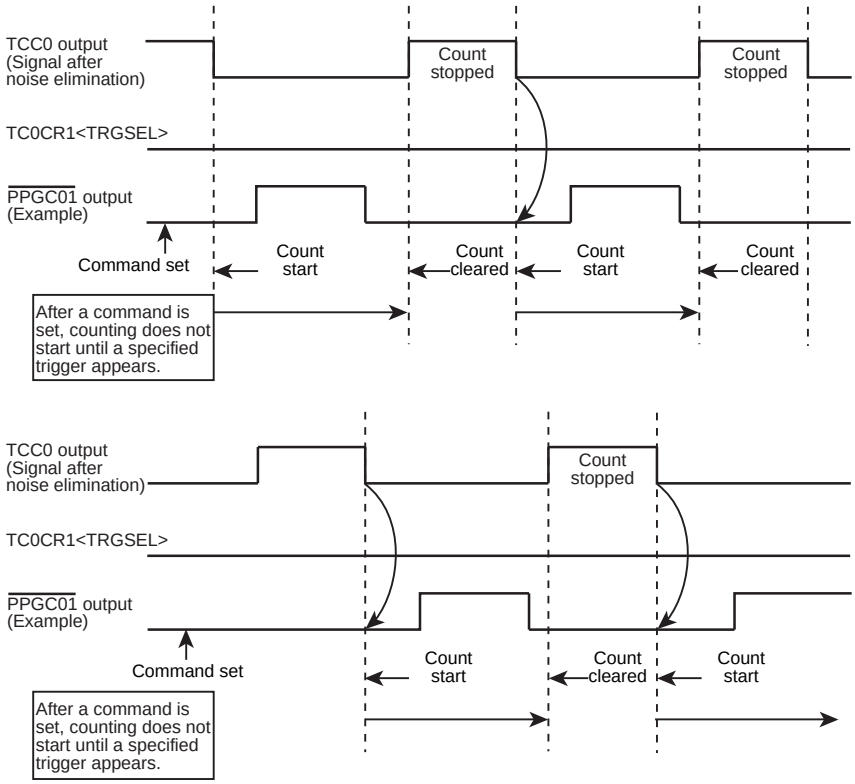


Figure 15-9 Example Operation in Trigger Start Mode

15.5.3 Trigger capture

15.5.3.1 Description

When counting starts in command start and capture mode, the count values at the rising and falling edges of the TCC0 pin input are captured and stored in capture registers TC0CAPA and TC0CAPB,respectively.

The captured data is first stored in the capture buffer. At the end of the period, the data is transferred from the capture buffer to the capture register. If a trigger input does not appear within a period, the data captured in the previous period remains in the capture buffer and is transferred to the capture register at the end of the period. If more than one trigger edge is detected within a period, the data captured last is written to the capture register.

Captured data must be read in the following order: Lower byte of capture register A (TC0CAPAL),upper byte of capture register A (TC0CAPAH), lower byte of capture register B (TC0CAPBL), and upper byte of capture register B (TC0CAPBH). Note that reading only the rising-edge captured data (TC0CAPA) does not update the next captured data. The falling-edge captured data (TC0CAPB) must also be read.

An attempt to read a captured value from a register other than the upper byte of the TC0CAPB causes the capture registers to enter protected state, in which captured data cannot be updated. Reading a value from the upper byte of the TC0CAPB cancels that state, re-enabling the updating of captured data (The TC0CAPA and TC0CAPB are read as a single set of operation).

Note that the protected state may be still effective immediately after the counter starts. Ensure that a dummy read of capture registers is performed in the first period to cancel the protected state.

The capture feature of the TCC0 assumes that a capture trigger (Rising or falling edge) appears within a period. Captured data is updated (An edge is detected) only when the timer is operating (TC0CR3<TCCST> = 1). If a timer stop command (TC0CR3<TCCST> = 0) is written within a period, captured data will be undefined. Captured data is not updated after a one-time stop command is written. In one-time stop mode, no trigger is accepted after a STOP command is given.

Downloaded from Arrow.com.

15.5.3.2 Register settings

- TC0CR2<CSTC> = “00” Command start and capture mode
- TC0CR3<STM> Continuous/one-time output
- TC0CR3<TCCST> = “1” Starts counting

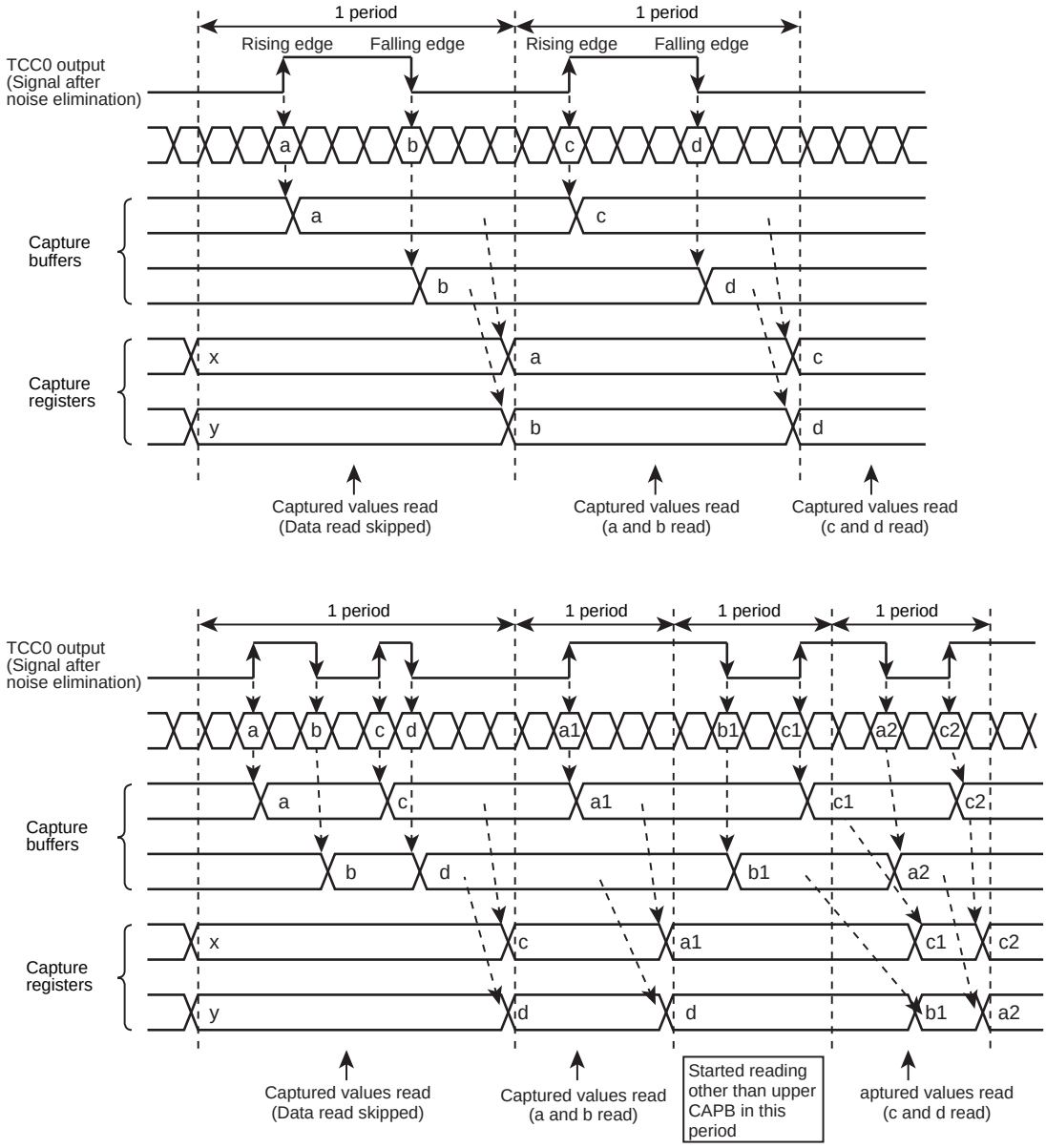


Figure 15-10 Example Operation in Trigger Capture Mode

15.5.4 Trigger start/stop acceptance mode

15.5.4.1 Selecting an input signal logic for the TCC0 pin (Trigger input)

The logic for an input trigger signal on the TCC0 pin can be specified using TC0CR1<TRGSEL> .

- TC0CR1<TRGSEL> = “0”: Counting starts on the falling edge. The counter is cleared and stopped while the TCC0 pin is high.

Downloaded from Arrow.com.

- TC0CR1<TRGSEL> = “1”: Counting starts on the rising edge. The counter is cleared and stopped while the TCC0 pin is low.

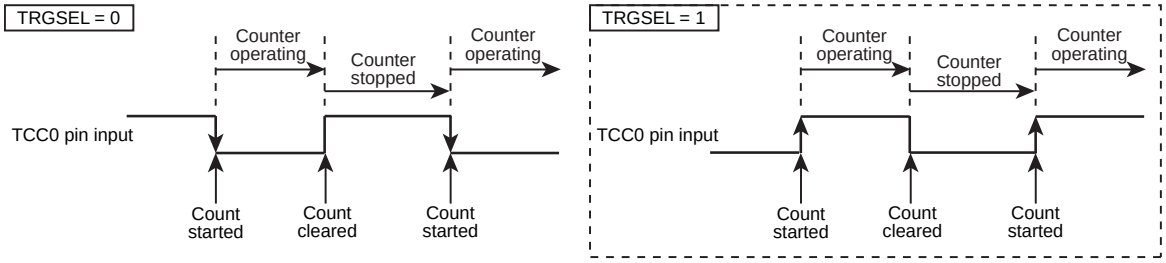
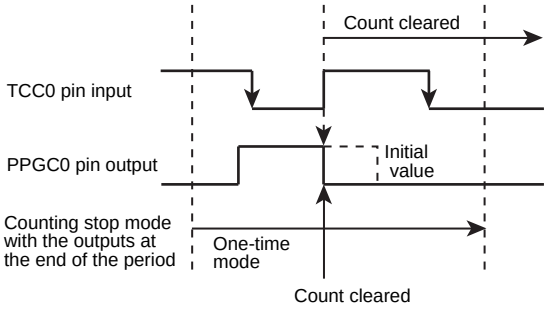


Figure 15-11 Trigger Input Signal

When TC0CR1<TRGSEL> is set to 0 to select a falling-edge trigger, a falling edge detected on the TCC0 pin causes the counter to start counting and a high level on the TCC0 pin causes the counter to be cleared and the PPG output to be initialized. The counter is stopped while the TCC0 pin input is high.

When TC0CR1<TRGSEL> is set to 1 to select a rising-edge trigger, a rising edge detected on the TCC0 pin causes the counter to start counting and a low level on the TCC0 pin causes the counter to be cleared and the PPG output to be initialized. The counter is stopped while the TCC0 pin input is low.

In one-time stop mode, the counter accepts a stop trigger but does not accept a start trigger (when a stop trigger is accepted within a period, the output is immediately initialized and the counter is stopped).



All triggers (Start and stop) are ignored when the timer is stopped (TC0CR3<TCCST> = 0).

15.5.4.2 Specifying whether triggers are always accepted or ignored when PPG outputs are active

The TC0CR1<TRGAM> specifies whether triggers from the TCC0 pin are always accepted or ignored when the PPG output is active.

- TC0CR1<TRGAM> = “0”

  Triggers from the TCC0 pin are always accepted regardless of whether $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs are active or inactive. A trigger starts or clears/stops the timer and deactivates $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs.

- TC0CR1<TRGAM> = “1”

  After TC0CR2<PPGCxOE> setting to "1" ,Triggers from the TCC0 pin are accepted only when $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs are inactive.A trigger starts or clears/stops the timer. Triggers are ignored when $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs are active.Tiggers are always accept the output of $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ when TC0CR2<PPGCxOE> is set in "0" (x = 1, 2).

Downloaded from Arrow.com.

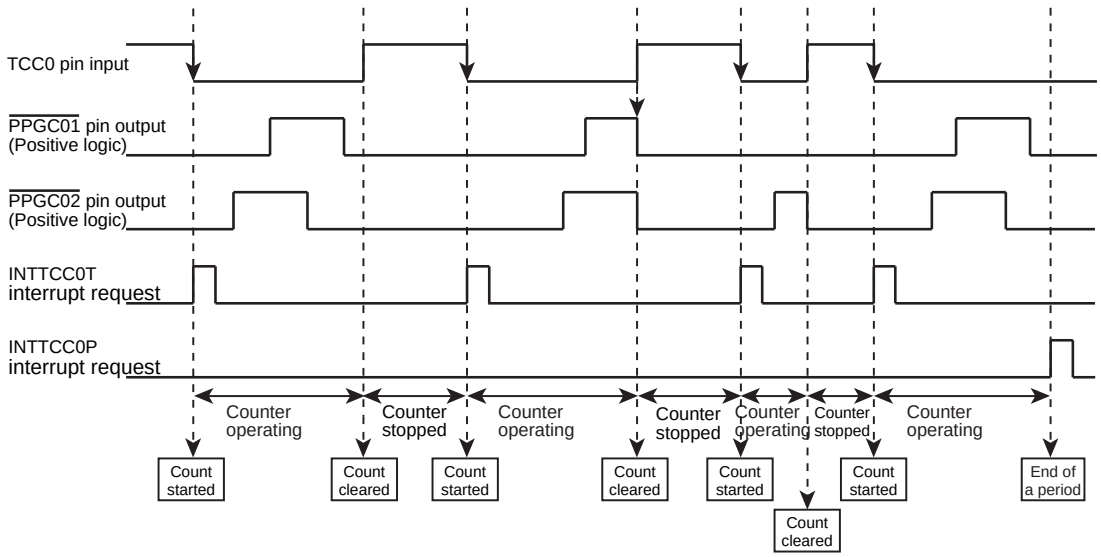


Figure 15-12 Start and Clear/stop Triggers on the TC7 Pin: Falling-edge Trigger (Counting stopped at high level), Triggers Always Accepted

15.5.4.3 Ignoring triggers when PPG outputs are active

Setting TRGAM to 1 specifies that triggers are ignored when PPG outputs are active; trigger edges detected when $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs are inactive are accepted and cause the counter to be cleared and stopped. If a trigger is detected when $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs are active, the counter does not stop immediately but continues counting until the outputs become inactive. If the trigger signal level is a stop level when the outputs become inactive, the counter is cleared/stopped and waits for a next start trigger. If output is enabled for both $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$, triggers are accepted only when both $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs are inactive.

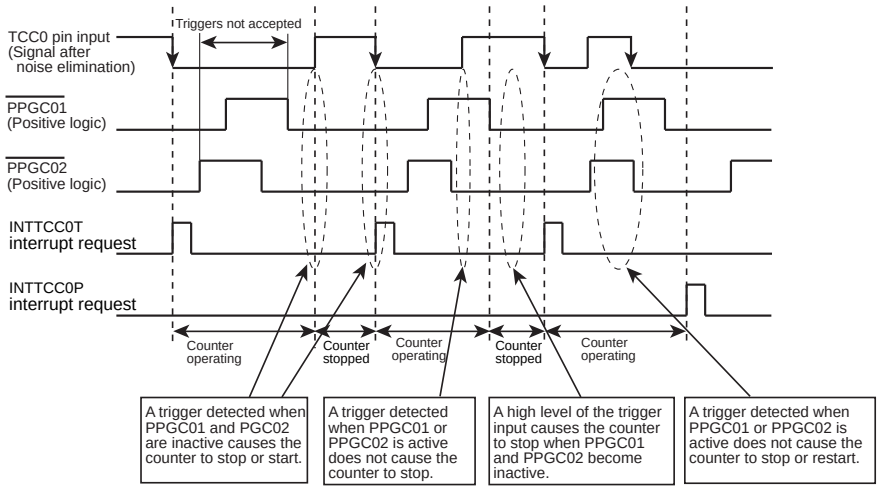


Figure 15-13 Start Triggers on the TCC0 Pin: Falling-edge Trigger (Counting stopped at high level), Triggers Ignored when PPG Outputs are Active

15.5.5 Configuring how the timer stops

Setting TC0CR3<TCCST> to 0 causes the timer to stop with the specified output state according to the setting of TC0CR3<STM>.

Downloaded from Arrow.com.

15.5.5.1 Counting stopped with the outputs initialized

When TC0CR3<STM> is set to 00, the counter stops immediately with the $\overline{PPGC01}$ and $\overline{PPGC02}$ outputs initialized to the values specified with PPGC1INI and PPGC2INI.

15.5.5.2 Counting stopped with the outputs maintained

When TC0CR3<STM> is set to 01, the counter stops immediately with the current $\overline{PPGC01}$ and $\overline{PPGC02}$ output states maintained

To restart the counter from the maintained state (TC0CR3<STM> = 01), set TC0CR3<TCCST> to 1. The counter is restarted with the initial output values, specified with PPGC1INI and PPGC2INI.

15.5.5.3 Counting stopped with the outputs initialized at the end of the period

When TC0CR3<STM> is set to 10, the counter continues counting until the end of the current period and then stops. If a stop trigger is detected before the end of the period, however, the counter stops immediately.

TC0CR1 and TC0CR2 must not be rewritten before the counter stops completely.

The TC0CR3<CNTBF> can be read to determine whether the counter has stopped.

15.5.6 One - time/continuous output mode

15.5.6.1 One-time output mode

Starting the timer (TC0CR3<TCCST> = 1) with TC0CR3<STM> set to 10 specifies one-time output mode. In this mode, the timer stops counting at the end of a period.

For a trigger start, the counter is stopped until a trigger is detected. A specified trigger restarts counting and the counter stops at the end of the period or when a stop trigger is detected, after which it waits for a trigger again.

For a command start, the counter is stopped until TC0CR3<TCCST> is reset to 1.

TC0CR1 and TC0CR2 must not be rewritten before the counter stops completely.

The TC0CR3<CNTBF> can be read to determine whether the counter has stopped.

TC0CR3<TCCST> remains set to 1 after the counter is stopped.

When TC0CR3<TCCST> is set to 1, setting TC0CR3<STM> to 10 clears the counter, which then restarts counting from the beginning in one-time output mode.

15.5.6.2 Continuous output mode

Starting the timer (TC0CR3<TCCST> = 1) with TC0CR3<STM> set to 00 or 01 specifies continuous output mode. In this mode, the timer outputs specified waveforms continuously.

Downloaded from Arrow.com.

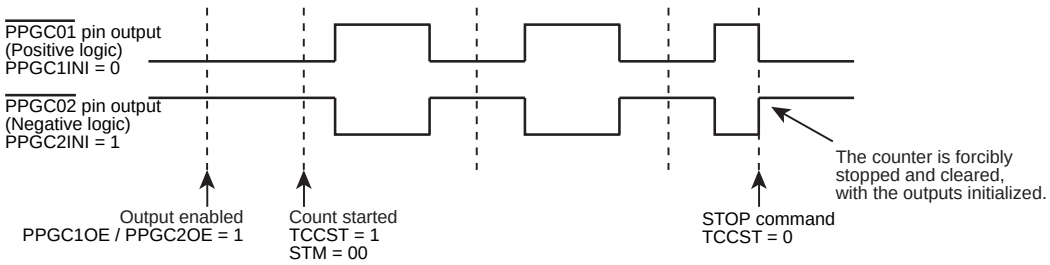


Figure 15-14 Immediately Stopping and Clearing the Counter with the Outputs Initialized (TC0CR3<STM> = “00”)

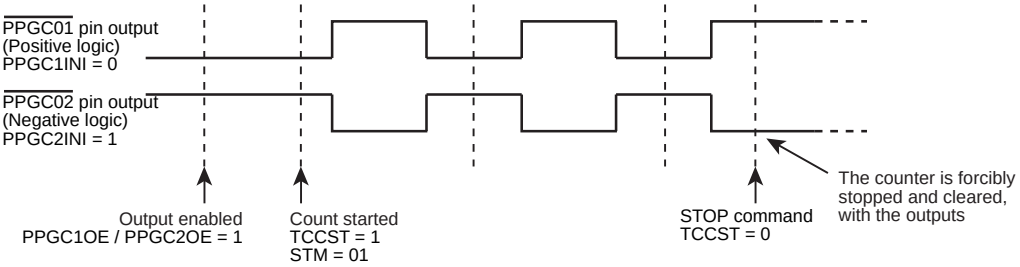


Figure 15-15 Immediately Stopping and Clearing the Counter with the Outputs Maintained (TC0CR3<STM> = “01”)

Downloaded from Arrow.com.

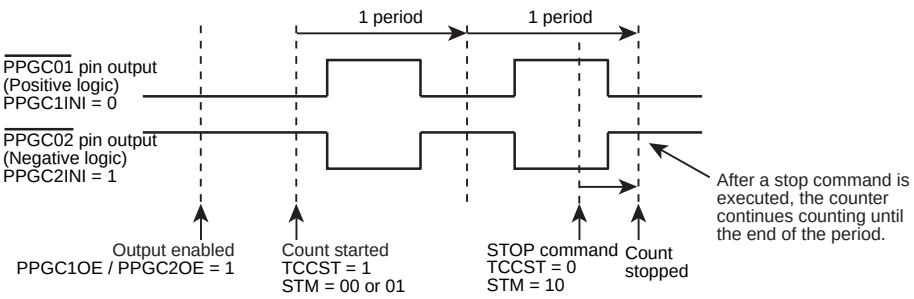


Figure 15-16 Stopping the Counter at the End of the Period (TC0CR3<STM> = “10”)

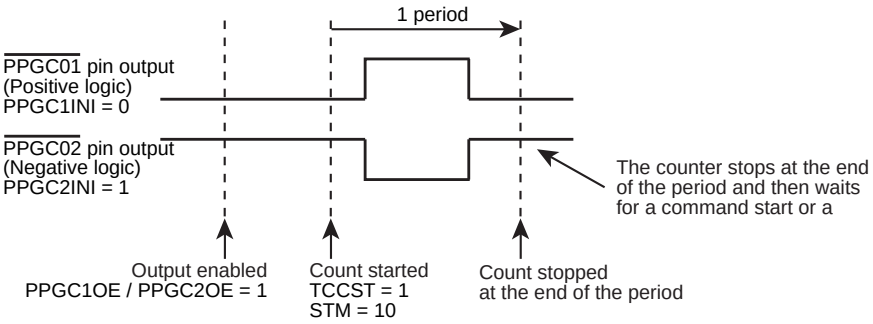


Figure 15-17 Stopping the Counter at the End of the Period (TC0CR3<STM> = “10”), TC0CR3<TCCST> = “1” One - time Output Mode

15.5.7 PPG output control (Initial value/output logic, enabling/disabling output)

15.5.7.1 Specifying initial values and output logic for PPG outputs

The TC0CR1<PPGC1INI and PPGC2INI> specify the initial values of $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ outputs as well as their output logic.

(1) Positive logic output

Setting the bit to 0 specifies that the output is initially low and driven high upon a match between the counter value and specified dead time.

(2) Negative logic output

Setting the bit to 1 specifies that the output is initially high and driven low upon a match between the counter value and specified dead time.

15.5.7.2 Enabling or disabling PPG outputs

The setting of the I/O port specify whether PPG outputs are enabled or disabled.When outputs are disabled, no PPG waveforms appear while the counter is operating, allowing the $\overline{\text{PPGC01}}$ and $\overline{\text{PPGC02}}$ pins to be used as normal input/output pins.

Downloaded from Arrow.com.

15.5.7.3 Using the TCC0 as a normal timer/counter

The TCC0 can be used as a normal timer/counter when PPG outputs are disabled using the setting of the I/O port.In that case, use an INTTCC0P interrupt, which occurs upon a match with the value specified in the data register (TC0DRC). Setting count start mode (TC0CR2<CSTC>) in command start and capture mode.

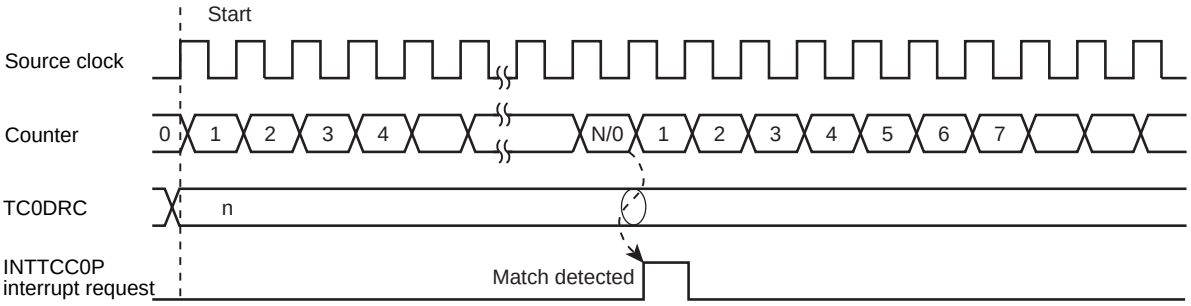


Figure 15-18 Using the TCC0 as a Normal Timer/Counter (When TC0CR3<CSIDIS>is "1")

15.5.8 Eliminating noise from the TCC0 pin input

A digital noise canceller eliminates noise from the input signal on the TCC0 pin.

The digital noise canceller uses a sampling clock of fcgck/4, fcgck/2 or fcgck, as specified with TC0CR1<NCRSEL>, and samples the signal five times. It accepts a level input which is continuous at least over the period of time required for five samplings. Any level input which does not continue over the period of time required for five samplings is canceled as noise.

Table 15-4 Noise Canceller Settings

TC0CR1 <NCRSEL>	Sampling Frequency (Number of Samplings)	Pulse Width Always Assumed as Noise			Pulse Width Always Assumed as Signal		
			at 8 MHz	at 16 MHz		at 8 MHz	at 16 MHz
00	fcgck/4 (5 times)	16/fcgck [s]	2 [μs]	1 [μs]	20/fcgck [s]	2.5 [μs]	1.25 [μs]
01	fcgck/2 (5 times)	8/fcgck [s]	1 [μs]	500 [ns]	10/fcgck [s]	1.25 [μs]	0.625 [μs]
10	fcgck (5 times)	4/fcgck [s]	0.5 [μs]	250 [ns]	5/fcgck [s]	0.625 [μs]	0.3125 [μs]
11	(None)	None	-	-	(1/fcgck)		

Downloaded from Arrow.com.

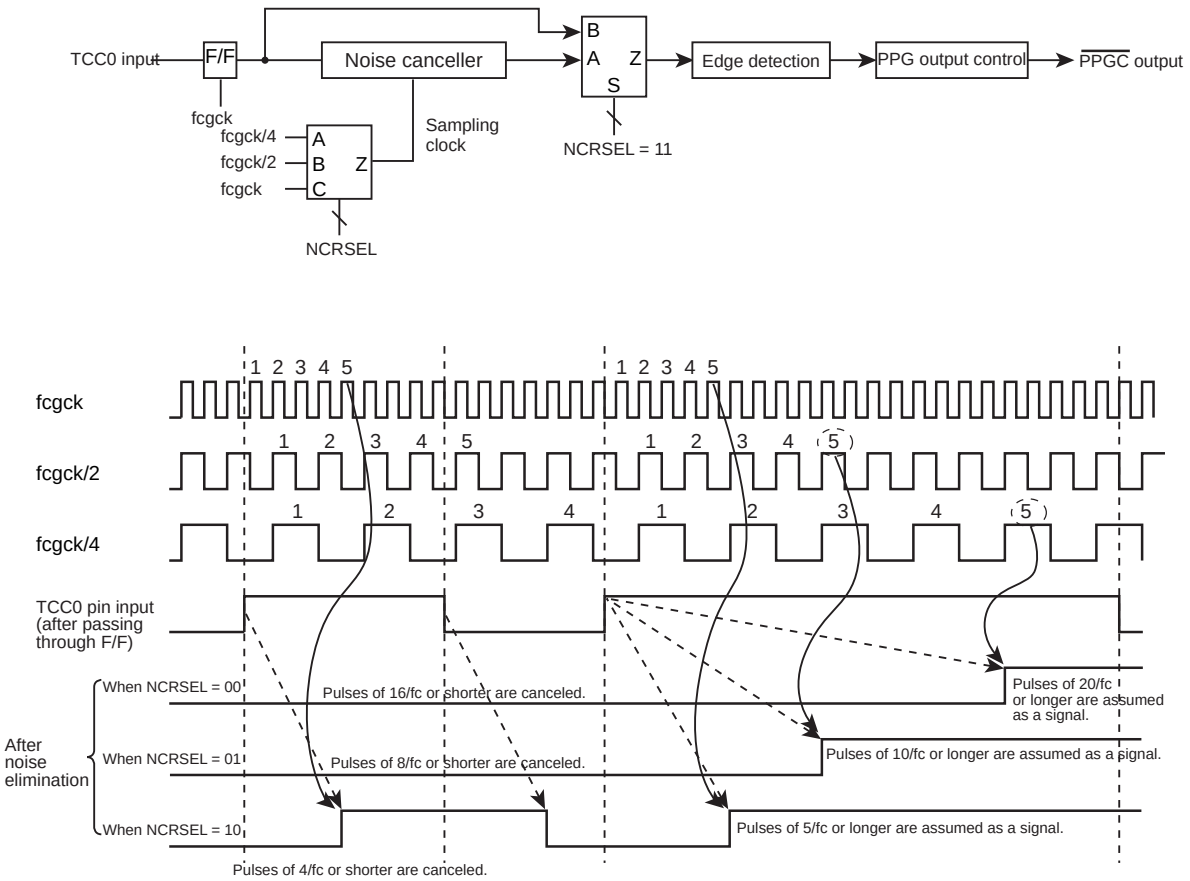


Figure 15-19 Noise Canceller Operation

- When TC0CR1<NCRSEL> = 00, a TCC0 input level after passing through the F/F is always canceled if its duration is 16/fcgck [s] or less and always assumed as a signal if its duration is 20/fcgck [s] or greater. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay between 21/fcgck [s] and 24/fcgck [s] before the PPG outputs vary.
- When TC0CR1<NCRSEL> = 01, a TCC0 input level after passing through the F/F is always canceled if its duration is 8/fcgck [s] or less and always assumed as a signal if its duration is 10/fcgck [s] or greater. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay between 13/fcgck [s] and 14/fcgck [s] before the PPG outputs vary.
- When TC0CR1<NCRSEL> = 10, a TCC0 input level after passing through the F/F is always canceled if its duration is 4/fcgck [s] or less and always assumed as a signal if its duration is 5/fcgck [s] or greater. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay of 5/fcgck [s] before the PPG outputs vary.
- When TC0CR1<NCRSEL> = 11, a pulse shorter than 1/fcgck may be assumed as a signal or canceled as noise in the first-stage F/F. Ensure that input signal pulses are longer than 1/fc. After the input signal supplied on the TCC0 pin passes through the F/F, there is a delay of 4/fcgck [s] before the PPG outputs vary.

Note 1: If the pin input level changes while the specified noise elimination threshold is being modified, the noise canceller may assume noise as a pulse or cancel a pulse as noise.

Note 2: If noise occurs in synchronization with the internal sampling timing consecutively, it may be assumed as a signal.

Note 3: The signal supplied on the TCC0 pin requires 1/fcgck [s] or less to pass through the F/F.

15.5.9 Interrupts

The TCC0 supports three interrupt sources.

Downloaded from Arrow.com.

15.5.9.1 INTTCC0T (Trigger start interrupt)

A trigger interrupt (INTTCC0T) occurs when the counter starts upon the detection of a trigger edge specified with TC0CR1<TRGST>. This interrupt does not occur with a trigger edge for clearing the count. A trigger edge detected in trigger capture mode does not cause an interrupt. A start trigger causes an interrupt even when the counter is stopped in emergency.

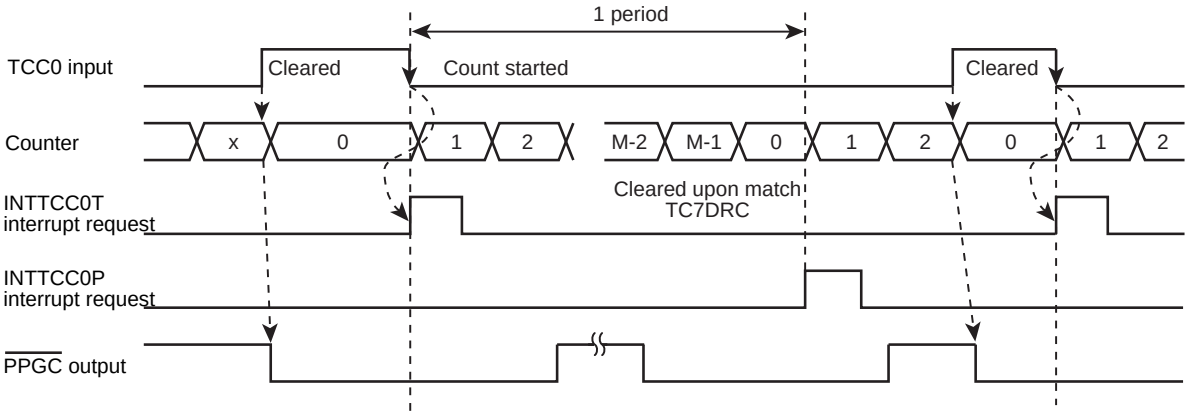


Figure 15-20 Trigger Start Interrupt

15.5.9.2 INTTCC0P (Period interrupt)

A period interrupt (INTTCC0P) occurs when the counter starts with a command and when the counter is cleared with the specified counter period (TC0DRC) reached, that is, at the end of a period. A match with the set period causes an interrupt even when the counter is stopped in emergency.

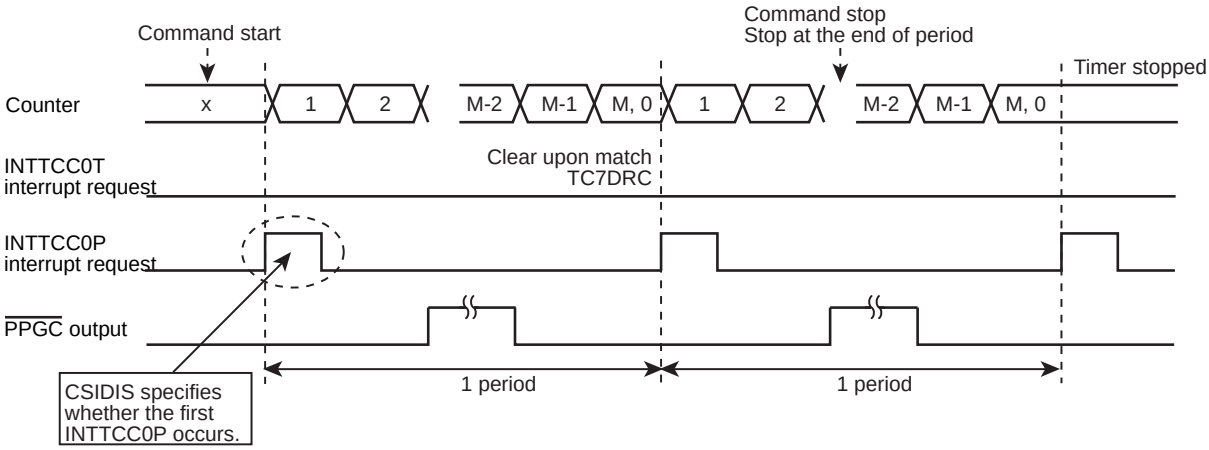


Figure 15-21 Period Interrupt

If a command start is specified (1 is written in TC0CR3<TCCST>) when the TCC0 pin is at a stop level, the counter does not start (INTTCC0P does not occur); a subsequent trigger start edge causes the counter to start and INTTCC0T to occur.

15.5.9.3 INTEMG0 (Emergency output stop interrupt)

An emergency output stop interrupt (INTEMG0) occurs when the emergency output stop circuit operates to stop PPG outputs in emergency

Downloaded from Arrow.com.

15.5.10 Emergency PPG output stop feature

Setting TC0CR2<EMGIE> to 1 enables the emergency PPG output stop feature (Enables the $\overline{\text{EMG0}}$ pin input).

A low level input detected on the $\overline{\text{EMG0}}$ pin causes an EMG interrupt (INTEMG0) to occur with the PPG waveforms initialized (as specified with PPGC1INI and PPGC2INI). (Emergency PPG output stop)

This feature only disables PPG outputs without stopping the counter. Use the EMG interrupt handler routine to stop the timer.

Note: Ensure that a low level on the $\overline{\text{EMG0}}$ pin continues for at least 4/fcgck [s]. The emergency PPG output stop feature may not operate normally with a low level shorter than 4/fcgck [s].

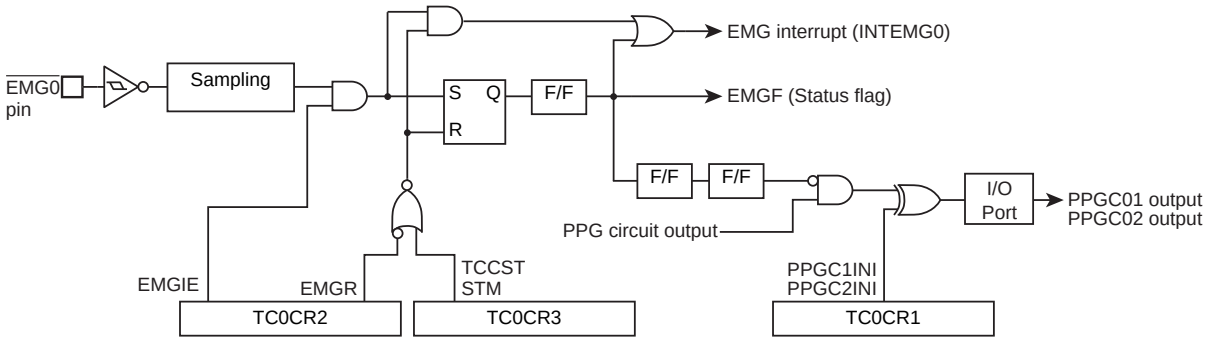


Figure 15-22 $\overline{\text{EMG0}}$ Pin

15.5.10.1 Enabling/disabling input on the $\overline{\text{EMG0}}$ pin

Setting TC0CR2<EMGIE> to 1 enables input on the $\overline{\text{EMG0}}$ pin and setting the bit to 0 disables input on the pin. (Initially, TC0CR2<EMGIE> is set to 0, disabling an emergency output stop ($\overline{\text{EMG0}}$ pin) input.)

The input signal on the $\overline{\text{EMG0}}$ pin is valid only when its shared port pin is placed in input mode. Ensure that the shared port pin is placed in input mode before attempting to enable the $\overline{\text{EMG0}}$ pin input.

The $\overline{\text{EMG0}}$ pin input is sampled using a high-frequency clock. The emergency PPG output stop feature does not operate normally if the high-frequency clock is stopped.

15.5.10.2 Monitoring the emergency PPG output stop state

When the emergency PPG output stop feature activates, the TC0CR3<EMGF> is set to 1. 1 read from EMGF indicates that PPG outputs are disabled by the emergency PPG output stop feature.

15.5.10.3 EMG interrupt

An EMG interrupt (INTEMG0) occurs when an emergency PPG output stop input is accepted. To use an INTEMG0 interrupt for some processing, ensure that the interrupt is enabled beforehand.

When the $\overline{\text{EMG0}}$ pin is low with TC0CT2<EMGIE> set to 1 ($\overline{\text{EMG0}}$ pin input enabled), an attempt to cancel the emergency PPG output stop state results in an interrupt being generated again, with the emergency PPG output stop state reestablished.

An INTEMG interrupt occurs whenever a stop input is accepted when TC0CR2<EMGIE> = 1, regardless of whether the timer is operating.

Downloaded from Arrow.com.

15.5.10.4 Canceling the emergency PPG output stop state

To cancel the emergency PPG output stop state, ensure that the input on the $\overline{\text{EMG0}}$ pin is high, set TC0CR3<TCCST> to 0 and TC0CR3<STM> to 00 to stop the timer, and then set TC0CR2<EMGR> to 1. Setting EMGR to 1 cancels the stop state only when TCCST = 0 and TC0CR3<STM> = 00; ensure that TC0CR3<TCCST> = 0 and TC0CR3<STM> = 00 before setting TC0CR2<EMGR> to 1.

If the input on the $\overline{\text{EMG0}}$ pin is low and TC0CR2<EMGIE> = 1 when the emergency PPG output stop state is canceled,the timer re-enters the emergency PPG output stop state and an INTEMG interrupt occurs.

15.5.10.5 Restarting the timer after canceling the emergency PPG output stop state

To restart the timer after canceling the emergency PPG output stop state, reconfigure the control registers (TC0CR1, TC0CR2, TC0CR3) before restarting the timer.

The timer cannot restart in the emergency PPG output stop state. Monitor the emergency PPG output stop state and cancel the state before reconfiguring the control registers to restart the timer. Ensure that the control registers are reconfigured according to the appropriate procedure for configuring timer operation control.

15.5.10.6 Response time between $\overline{\text{EMG0}}$ pin input and PPG outputs being initialized

The time between a low level input being detected on the $\overline{\text{EMG0}}$ pin and the PPG outputs being initialized is up to 4/fcgck [s].

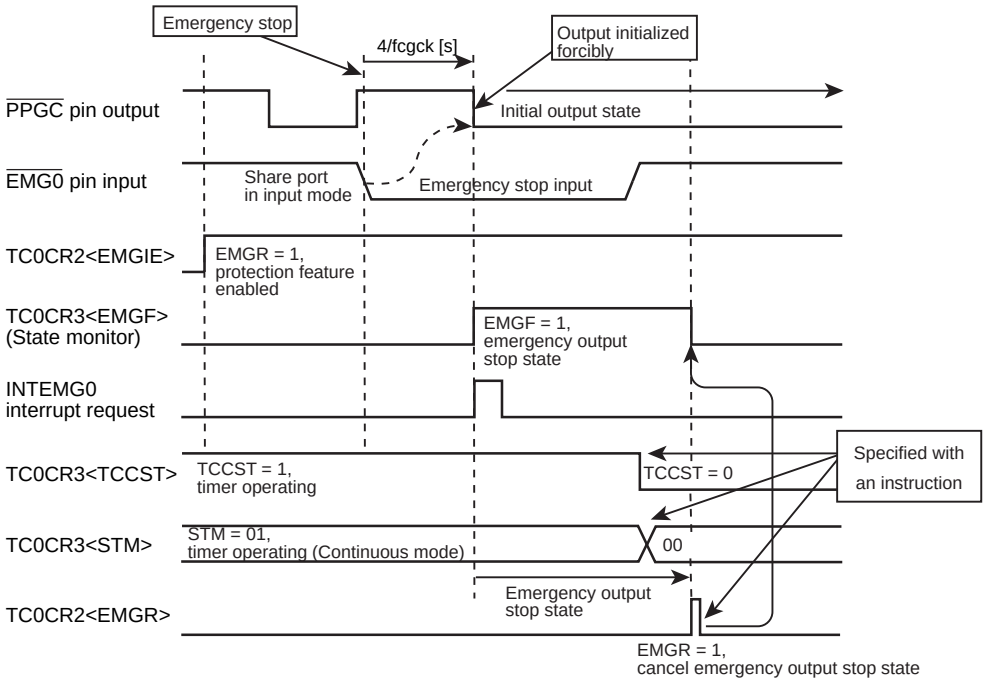


Figure 15-23 Timing between $\overline{\text{EMG0}}$ Pin Input being Detected and PPG Outputs being Disabled

15.5.11 TCC0 operation and microcontroller operating mode

The TCC0 operates when the microcontroller is placed in NORMAL1, NORMAL2, IDLE1, or IDLE2 mode. If the mode changes from NORMAL or IDLE to STOP, SLOW, or SLEEP while the TCC0 is operating, the TCC0 is initialized and stops operating.

Downloaded from Arrow.com.

To change the microcontroller operating mode from NORMAL to STOP, SLOW, or SLEEP, ensure that the TCC0 timer is stopped before attempting to execute a mode change instruction.

To change the mode from STOP, SLOW, or SLEEP to NORMAL to restart the TCC0, reconfigure all registers according to the appropriate TCC0 operation procedure.

15.5.12 Considerations for Using the development tools of TCC0

When stopped a program by the break of the debugger, TCC0 suspend PPG output,and change to initial value depend on TC0CR1<PPGC1INI and PPGC2INI> setting. When the break is released, PPG output is restarted from the point at which it was suspended. The setting of development tools can continue PPG output during break. For detail refer to the instruction manual of the development tools.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

16. 8-bit Timer Counter (TC0)

The TMP89FW24A contains 4 channels of high-performance 8-bit timer counters (TC0). Each timer can be used for time measurement and pulse output with a prescribed width. Two 8-bit timer counters are cascadable to form a 16-bit timer.

This chapter describes 2 channels of 8-bit timer counters 00 and 01. For 8-bit timer counters 02 and 03, replace the SFR addresses and pin names as shown in Table 16-1 and Table 16-2.

Table 16-1 SFR Address Assignment

	16-bit mode	T0xREG (Address)	T0xPWM (Address)	T0xMOD (Address)	T0xxCR (Address)	Low power consumption register
Timer counter 00	Lower	T00REG (0x00026)	T00PWM (0x00028)	T00MOD (0x0002A)	T001CR (0x0002C)	POFFCR0 <TC001EN>
Timer counter 01	Higher	T01REG (0x00027)	T01PWM (0x00029)	T01MOD (0x0002B)		
Timer counter 02	Lower	T02REG (0x00F88)	T02PWM (0x00F8A)	T02MOD (0x00F8C)	T023CR (0x00F8E)	POFFCR0 <TC023EN>
Timer counter 03	Higher	T03REG (0x00F89)	T03PWM (0x00F8B)	T03MOD (0x00F8D)		

Table 16-2 Pin Names

	Timer input pin	PWM output pin	PPG output pin
Timer counter 00	TC00 pin	$\overline{\text{PWM0}}$ pin	$\overline{\text{PPG0}}$ pin
Timer counter 01	TC01 pin	$\overline{\text{PWM1}}$ pin	$\overline{\text{PPG1}}$ pin
Timer counter 02	TC02 pin	$\overline{\text{PWM2}}$ pin	$\overline{\text{PPG2}}$ pin
Timer counter 03	TC03 pin	$\overline{\text{PWM3}}$ pin	$\overline{\text{PPG3}}$ pin

Downloaded from Arrow.com.

16.1 Configuration

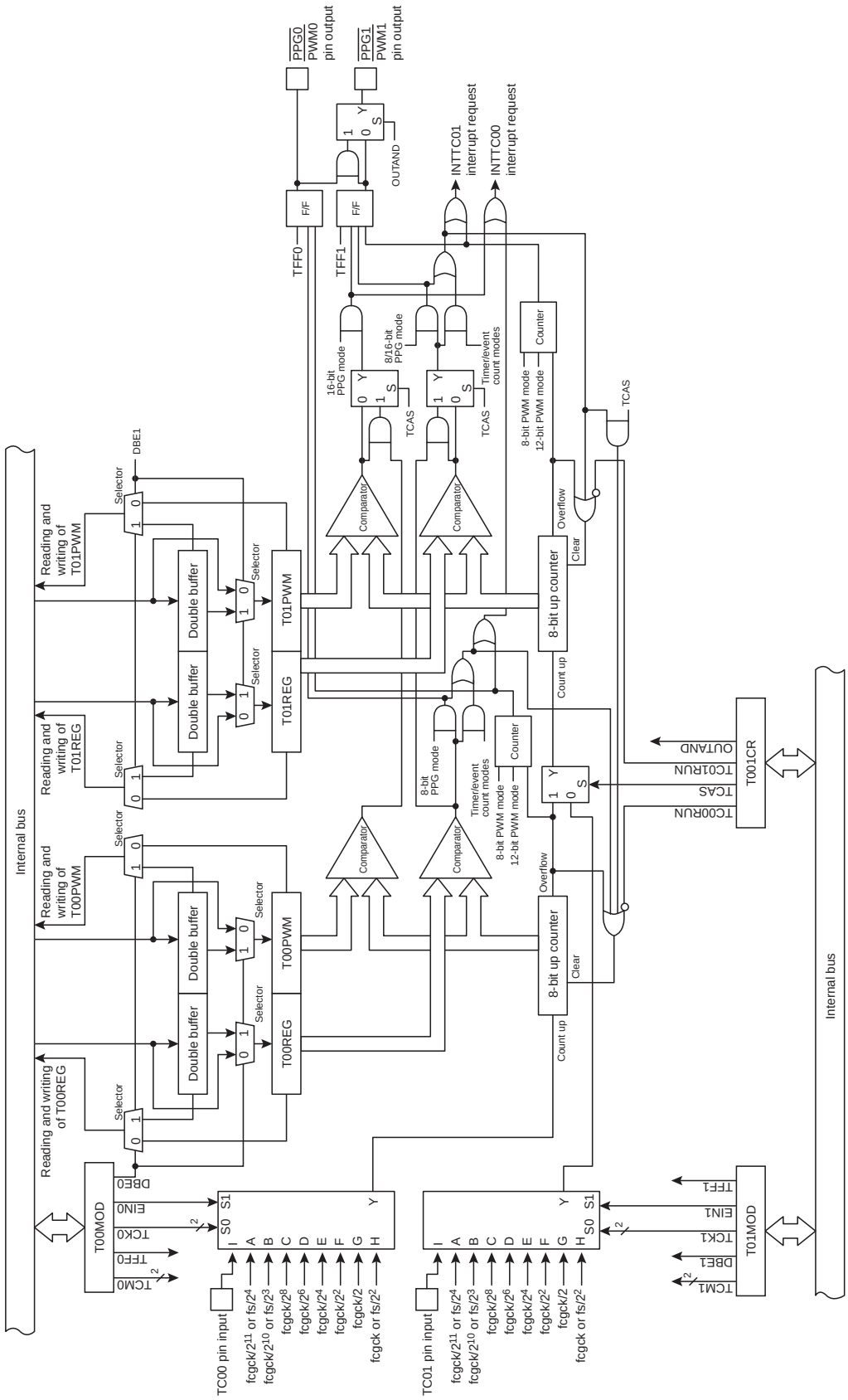


Figure 16-1 8-bit Timer Counters 00 and 01

Downloaded from Arrow.com.

16.2 Control

16.2.1 Timer counter 00

The timer counter 00 is controlled by the timer counter 00 mode register (T00MOD) and two 8-bit timer registers (T00REG and T00PWM).

Timer register 00

T00REG (0x00026)	15	14	13	12	11	10	9	8
Bit Symbol	T00REG							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer register 00

T00PWM (0x00028)	7	6	5	4	3	2	1	0
Bit Symbol	T00PWM							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Note 1: For the configuration of T00PWM in the 8-bit and 12-bit PWM modes, refer to "16.4.3 8-bit pulse width modulation (PWM) output mode" and "16.4.7 12-bit pulse width modulation (PWM) output mode".

Downloaded from Arrow.com.

Timer counter 00 mode register

T00MOD		7	6	5	4	3	2	1	0
(0x0002A)	Bit Symbol	TFF0	DBE0	TCK0			EIN0	TCM0	
	Read/Write	R/W	R/W	R/W			R/W	R/W	
	After reset	1	1	0	0	0	0	0	0

TFF0	Timer F/F0 control	0	Clear		
		1	Set		
DBE0	Double buffer control	0	Disable the double buffer		
		1	Enable the double buffer		
TCK0	Operation clock selection		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"	
		000	fcgck/2^{11}	fs/2^4	fs/2^4
		001	fcgck/2^{10}	fs/2^3	fs/2^3
		010	fcgck/2^8	fcgck/2^8	-
		011	fcgck/2^6	fcgck/2^6	-
		100	fcgck/2^4	fcgck/2^4	-
		101	fcgck/2^2	fcgck/2^2	-
		110	fcgck/2	fcgck/2	-
		111	fcgck	fcgck	fs/2^2
EIN0	Selection for using external source clock	0	Select the internal clock as the source clock.		
		1	Select an external clock as the source clock. (the falling edge of the TC00 pin)		
TCM0	Operation mode selection	00	8-bit timer/event counter modes		
		01	8-bit timer/event counter modes		
		10	8-bit pulse width modulation output (PWM) mode		
		11	8-bit programmable pulse generate (PPG) mode		

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Set T00MOD while the timer is stopped. Writing data into T00MOD is invalid during the timer operation.

Note 3: In the 8-bit timer/event modes, the TFF0 setting is invalid. In this mode, when the $\overline{PWM0}$ and $\overline{PPG0}$ pins are set as the function output pins in the port setting, the pins always output the "H" level.

Note 4: When EIN0 is set to "1" and the external clock input is selected as the source clock, the TCK0 setting is ignored.

Note 5: When the T001CR<TCAS> bit is "1", timer 00 operates in the 16-bit mode. The T00MOD setting is invalid and timer 00 cannot be used independently in this mode. When the $\overline{PWM0}$ and $\overline{PPG0}$ pins are set to the function output pins in the port setting, the pins always output the "H" level.

Note 6: When the 16-bit mode is selected at T001CR<TCAS>, the timer start is controlled at T001CR<T01RUN>. Timer 00 is not started by writing data into T001CR<T00RUN>.

Downloaded from Arrow.com.

16.2.2 Timer counter 01

Timer counter 01 is controlled by timer counter 01 mode register (T01MOD) and two 8-bit timer registers (T01REG and T01PWM).

Timer register 01

T01REG (0x00027)	15	14	13	12	11	10	9	8
Bit Symbol	T01REG							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM (0x00029)	7	6	5	4	3	2	1	0
Bit Symbol	T01PWM							
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

Note 1: For the configuration of T00PWM in the 8-bit and 12-bit PWM modes, refer to "16.4.3 8-bit pulse width modulation (PWM) output mode" and "16.4.7 12-bit pulse width modulation (PWM) output mode".

Downloaded from Arrow.com.

Timer counter 01 mode register

T01MOD (0x0002B)		7	6	5	4	3	2	1	0
	Bit Symbol	TFF1	DBE1	TCK1			EIN1	TCM1	
	Read/Write	R/W	R/W	R/W			R/W	R/W	
	After reset	1	1	0	0	0	0	0	0

TFF1	Timer F/F1 control	0	Clear		
		1	Set		
DBE1	Double buffer control	0	Disable the double buffer		
		1	Enable the double buffer		
TCK1	Operation clock selection		NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode
			SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"	
		000	fcgck/2^{11}	fs/2^4	fs/2^4
		001	fcgck/2^{10}	fs/2^3	fs/2^3
		010	fcgck/2^8	fcgck/2^8	-
		011	fcgck/2^6	fcgck/2^6	-
		100	fcgck/2^4	fcgck/2^4	-
		101	fcgck/2^2	fcgck/2^2	-
		110	fcgck/2	fcgck/2	-
		111	fcgck	fcgck	fs/2^2
EIN1	Selection for using external source clock	0	Select the internal clock as the source clock.		
		1	Select an external clock as the source clock. (the falling edge of the TC01 pin)		
TCM1	Operation mode selection		T001CR<TCAS>="0" (8-bit mode)		T001CR<TCAS>="1" (16-bit mode)
		00	8-bit timer/event counter modes		16-bit timer/event counter modes
		01	8-bit timer/event counter modes		16-bit timer/event counter modes
		10	8-bit pulse width modulation output (PWM) mode		12-bit pulse width modulation output (PWM) mode
		11	8-bit programmable pulse generate (PPG) mode		16-bit programmable pulse generate (PPG) mode

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: Set T01MOD while the timer is stopped. Writing data into T01MOD is invalid during the timer operation.

Note 3: In the 8-bit timer/event modes, the TFF1 setting is invalid. In this mode, when the $\overline{PWM1}$ and $\overline{PPG1}$ pins are set as the function output pins in the port setting, the pins always output the "H" level.

Note 4: When EIN1 is set to "1" and the external clock input is selected as the source clock, the TCK1 setting is ignored.

Downloaded from Arrow.com.

16.2.3 Common to timer counters 00 and 01

Timer counters 00 and 01 have the low power consumption register (POFFCR0) and timer 00 and 01 control registers in common.

Low power consumption register 0

POFFCR0 (0x00F74)	7	6	5	4	3	2	1	0
Bit Symbol	-	-	TC023EN	TC001EN	-	TCC0EN	TCB0EN	TCA0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

TC023EN	TC02,03 Control	0 1	Disable Enable
TC001EN	TC00,01 Control	0 1	Disable Enable
TCC0EN	TCC0 Control	0 1	Disable Enable
TCB0EN	TCB0 Control	0 1	Disable Enable
TCA0EN	TCA0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

Timer counter 01 control register

T001CR (0x0002C)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	OUTAND	TCAS	T01RUN	T00RUN
	Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

OUTAND	Timers 00 and 01 output control	0	Output the timer 00 output from the $\overline{PWM0}$ and $\overline{PPG0}$ pins and the timer 01 output from the $\overline{PWM1}$ and $\overline{PPG1}$ pins.
		1	Output a pulse that is a logical ANDed product of the outputs of timers 00 and 01 from the $\overline{PWM1}$ and $\overline{PPG1}$ pins.
TCAS	Timers 00 and 01 cascade control	0	Use timers 00 and 01 independently (8-bit mode).
		1	Cascade timers 00 and 01 (16-bit mode).
T01RUN	Timer 01 control Timers 00/01 control (16-bit mode)	0	Stop and clear the counter
		1	Start
T00RUN	Timer 00 control	0	Stop and clear the counter
		1	Start

Note 1: When STOP mode is started, T00RUN and T01RUN are cleared to "0" and the timers stop. Set T001CR again to use timers 00 and 01 after STOP mode is released.

Note 2: When a read instruction is executed on T001CR, bits 7 to 4 are read as "0".

Note 3: When OUTAND is "1", output is obtained from the $\overline{PWM1}$ and $\overline{PPG1}$ pins only. There is no timer output to the $\overline{PWM0}$ and $\overline{PPG0}$ pins. If the PWM0 and PPG0 pins are set as the function output pins in the port setting, the pins always output "H".

Note 4: OUTAND and TCAS can be changed only when both TC01RUN and TC00RUN are "0". When either TC01RUN or TC00RUN is "1" or both are "1", the register values remain unchanged by executing write instructions on OUTAND and TCAS. OUTAND and TCAS can be changed at the same time as TC01RUN and TC00RUN are changed from "0" to "1".

Downloaded from Arrow.com.

16.2.4 Operation modes and usable source clocks

The operations modes of the 8-bit timers and the usable source clocks are listed below.

Table 16-3 Operation Modes and Usable Source Clocks (NORMAL1/2 and IDLE1/2 modes)

TCK0		000	001	010	011	100	101	110	111	TC0i pin input
Operation mode		fcgck/2^{11} or fs/2^4	fcgck/2^{10} or fs/2^3	fcgck/2^8	fcgck/2^6	fcgck/2^4	fcgck/2^2	fcgck/2	fcgck	
8-bit timer modes	8-bit timer	O	O	O	O	O	O	O	O	-
	8-bit event counter	-	-	-	-	-	-	-	-	O
	8-bit PWM	O	O	O	O	O	O	O	O	-
	8-bit PPG	O	O	O	O	O	O	O	O	-
16-bit timer modes	16-bit timer	O	O	O	O	O	O	O	O	-
	16-bit event counter	-	-	-	-	-	-	-	-	O
	12-bit PWM	O	O	O	O	O	O	O	O	O
	16-bit PPG	O	O	O	O	O	O	O	O	O

Note 1: O: Usable, -: Unusable

Note 2: Set the source clock in the 16-bit modes on the TC01 side (TCK1).

Note 3: When the low-frequency clock, fs, is not oscillating, it must not be selected as the source clock. If fs is selected when it is not oscillating, no source clock is supplied to the timer, and the timer remains stopped.

Note 4: i=0, 1 (i=0 only in the 16-bit modes)

Note 5: The operation modes of the 8-bit timers and the usable source clocks are listed below.

Table 16-4 Operation Modes and Usable Source Clocks (SLOW1/2 and SLEEP1 modes)

TCK0		000	001	010	011	100	101	110	111	TC0i pin input
Operation mode		fs/2^4	fs/2^3	-	-	-	-	-	fs/2^2	
8-bit timer modes	8-bit timer	O	O	-	-	-	-	-	O	-
	8-bit event counter	-	-	-	-	-	-	-	-	O
	8-bit PWM	O	O	-	-	-	-	-	O	-
	8-bit PPG	O	O	-	-	-	-	-	O	-
16-bit timer modes	16-bit timer	O	O	-	-	-	-	-	O	-
	16-bit event counter	-	-	-	-	-	-	-	-	O
	12-bit PWM	O	O	-	-	-	-	-	O	O
	16-bit PPG	O	O	-	-	-	-	-	O	O

Note 1: O: Usable, -: Unusable

Note 2: Set the source clock in the 16-bit modes on the TC01 side (TCK1).

Note 3: i=0, 1 (i=0 only in the 16-bit modes)

Downloaded from Arrow.com.

16.3 Low Power Consumption Function

Timer counters 00 and 01 have the low power consumption registers (POFFCR0) that save power when the timers are not used.

Setting POFFCR0<TC001EN> to "0" disables the basic clock supply to timer counters 00 and 01 to save power. Note that this renders the timers unusable. Setting POFFCR0<TC001EN> to "1" enables the basic clock supply to timer counters 00 and 01 and allows the timers to operate.

After reset, POFFCR0<TC001EN> are initialized to "0", and this makes the timers unusable. When using the timers for the first time, be sure to set POFFCR0<TC001EN> to "1" in the initial setting of the program (before the timer control registers are operated).

Do not change POFFCR0<TC001EN> to "0" during the timer operation. Otherwise timer counters 00 and 01 may operate unexpectedly.

Downloaded from Arrow.com.

16.4 Functions

Timer counters TC00 and TC01 have 8-bit modes in which they are used independently and 16-bit modes in which they are cascaded.

The 8-bit modes include four operation modes; the 8-bit timer mode, the 8-bit event counter mode, the 8-bit pulse width modulation output (PWM) mode and the 8-bit programmable pulse generated output (PPG) mode.

The 16-bit modes include four operation modes; the 16-bit timer mode, the 16-bit event counter mode, the 12-bit PWM mode and the 16-bit PPG mode.

16.4.1 8-bit timer mode

In the 8-bit timer mode, the up-counter counts up using the internal clock, and interrupts can be generated regularly at specified times. The operation of TC00 is described below, and the same applies to the operation of TC01. (Replace TC00- by TC01-).

16.4.1.1 Setting

TC00 is put into the 8-bit timer mode by setting T00MOD<TCM0> to "00" or "01", T001CR<TCAS> to "0" and T00MOD<EIN0> to "0". Select the source clock at T00MOD<TCK0>. Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

16.4.1.2 Operation

Setting T001CR<T00RUN> to "1" allows the 8-bit up counter to increment based on the selected internal source clock. When a match between the up counter value and the T00REG set value is detected, an INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Setting T001CR<T00RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x00".

16.4.1.3 Double buffer

The double buffer can be used for T00REG by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T00REG during the timer operation, the set value is initially stored in the double buffer, and T00REG is not immediately updated. T00REG compares the previous set value with the up counter value. When the values match, an INTTC00 interrupt request is generated and the double buffer set value is stored in T00REG. Subsequently, the match detection is executed using a new set value.

  When a write instruction is executed on T00REG while the timer is stopped, the set value is immediately stored in both the double buffer and T00REG.

- When the double buffer is disabled

  When a write instruction is executed on T00REG during the timer operation, the set value is immediately stored in T00REG. Subsequently, the match detection is executed using a new set value.

  If the value set to T00REG is smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request interval may be longer than the selected time. If the value set to T00REG is equal to the up coun-

Downloaded from Arrow.com.

ter value, the match detection is executed immediately after data is written into T00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock (Figure 16-3). If these are problems, enable the double buffer.

When a write instruction is executed on T00REG while the timer is stopped, the set value is immediately stored in T00REG.

When a read instruction is executed on T00REG, the last value written into T00REG is read out, regardless of the T00MOD<DBE0> setting.

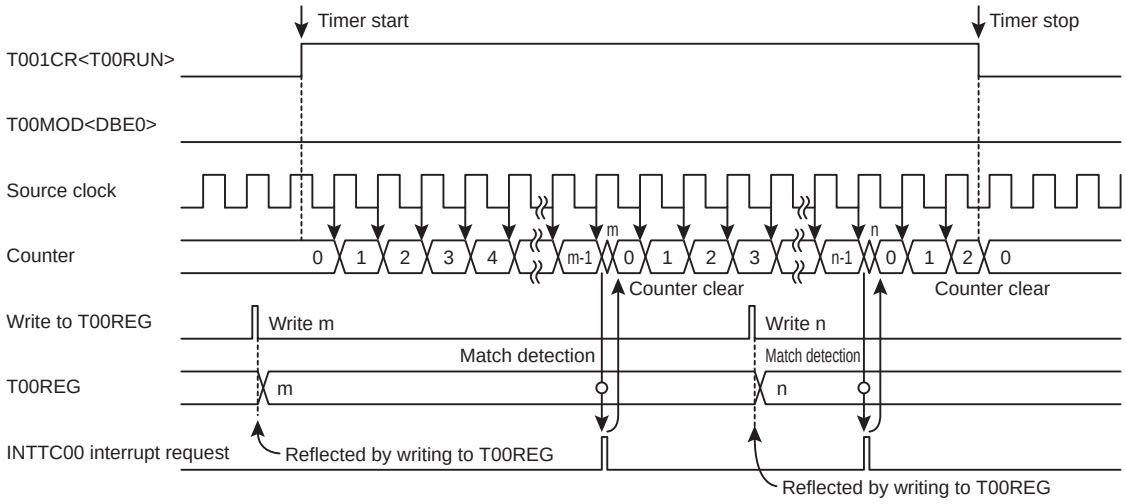
Table 16-5 8-bit Timer Mode Resolution and Maximum Time Setting

T00MOD <TCK0>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	fcgck/2^{11}	fs/2^4	fs/2^4	204.8μs	488.2μs	52.2ms	124.5ms
001	fcgck/2^{10}	fs/2^3	fs/2^3	102.4μs	244.1μs	26.1ms	62.3ms
010	fcgck/2^8	fcgck/2^8	-	25.6μs	-	6.5ms	-
011	fcgck/2^6	fcgck/2^6	-	6.4μs	-	1.6ms	-
100	fcgck/2^4	fcgck/2^4	-	1.6μs	-	408μs	-
101	fcgck/2^2	fcgck/2^2	-	400ns	-	102μs	-
110	fcgck/2	fcgck/2	-	200ns	-	51μs	-
111	fcgck	fcgck	fs/2^2	100ns	122.1μs	25.5μs	31.1ms

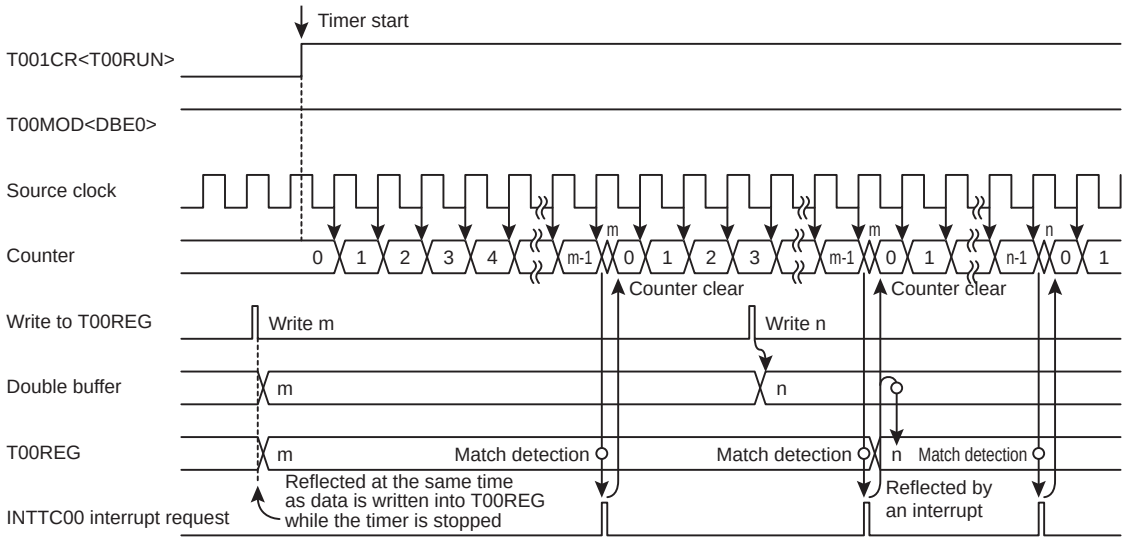
(Example) Operate TC00 in the 8-bit timer mode with the operation clock of fcgck/2^2 [Hz] and generate interrupts at 64 μs intervals (fcgck = 10 MHz)

```
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xE8       ; Selects the 8-bit timer mode and fcgck/2^2
LD      (T00REG),0xA0       ; Sets the timer register (64μs / (2^2/fcgck) = 0xA0)
SET     (T001CR).0          ; Starts TC00
```


2026-04-21
Downloaded from Arrow.com.



When the double buffer is disabled (T00MOD<DBE0>="0")



When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 16-2 Timer Mode Timing Chart

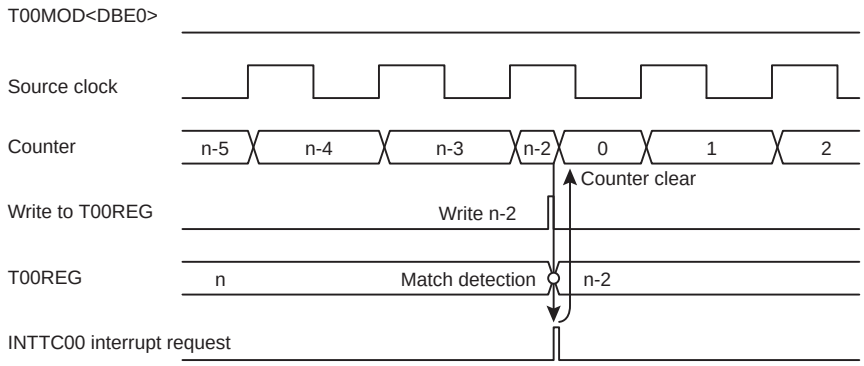


Figure 16-3 Operation When T00REG and the Up Counter Have the Same Value

Downloaded from Arrow.com.

16.4.2 8-bit event counter mode

In the 8-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 or TC01 pin. The operation of TC00 is described below, and the same applies to the operation of TC01.

16.4.2.1 Setting

TC00 is put into the 8-bit event counter mode by setting T00MOD<TCM0> to "00", T001CR<TCAS> to "0" and T00MOD<EIN0> to "1". Set the count value to be used for the match detection as an 8-bit value at the timer register T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

16.4.2.2 Operation

Setting T001CR<T00RUN> to "1" allows the 8-bit up counter to increment at the falling edge of the TC00 pin. When a match between the up-counter value and the T00REG set value is detected, an INTTC00 interrupt request is generated and the up counter is cleared to "0x00". After being cleared, the up counter restarts counting. Setting T001CR<T00RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x00".

The maximum frequency to be supplied is $fcgck/2^2$ [Hz] (in NORMAL1/2 or IDLE1/2 mode) or fs/24 [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

16.4.2.3 Double buffer

Refer to "16.4.1.3 Double buffer".

(Example) Operate TC00 in the 8-bit event counter mode and generate an interrupt each time 16 falling edges are detected at the TC00 pin.

```
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xC4       ; Selects to the 8-bit event counter mode
LD      (T00REG),0x10       ; Sets the timer register
SET     (T001CR).0          ; Starts TC00
```


Downloaded from Arrow.com.

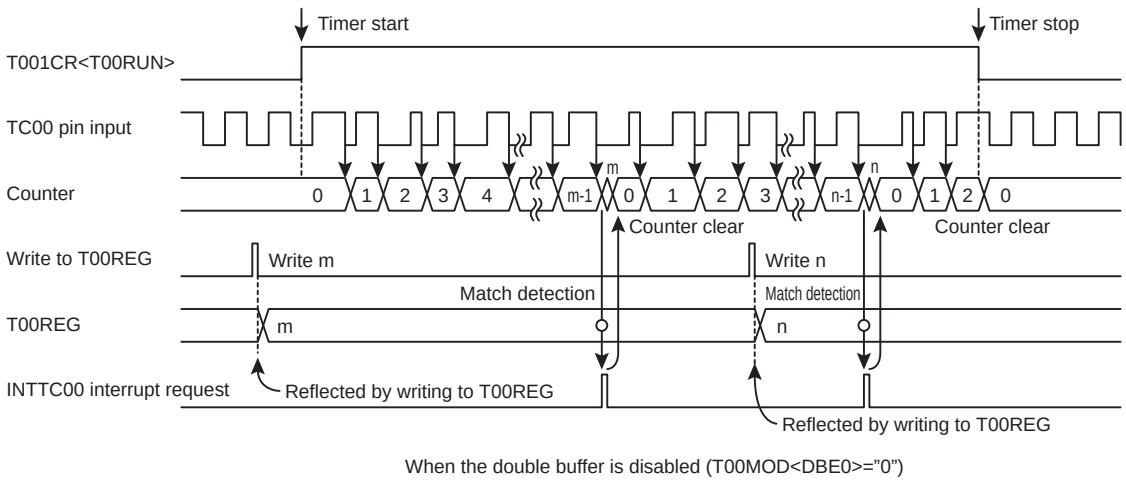


Figure 16-4 Event Counter Mode Timing Chart

Downloaded from Arrow.com.

16.4.3 8-bit pulse width modulation (PWM) output mode

The pulse-width modulated pulses with a resolution of 7 bits are output in the 8-bit PWM mode. An additional pulse can be added to the 2 × n-th duty pulse. This enables PWM output with a resolution nearly equivalent to 8 bits. (n=1, 2, 3...)

The operation of TC00 is described below, and the same applies to the operation of TC01.

16.4.3.1 Setting

TC00 is put into the 8-bit PWM mode by setting T00MOD<TCM0> to "10" and T001CR<TCAS> to "0". Set T00MOD<EIN0> to "0" and select the clock at T00MOD<TCK0>. Set the count value to be used for the match detection and the additional pulse value at the PWM register T00PWM.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

In the 8-bit PWM mode, the T00PWM register is configured as follows:

Timer register 00

T00PWM		7	6	5	4	3	2	1	0
(0x00028)	Bit Symbol	PWMDUTY							PWMAD
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM		7	6	5	4	3	2	1	0
(0x00029)	Bit Symbol	PWMDUTY							PWMAD
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

PWMDUTY is a 7-bit register used to set the duty pulse width value (the time before the first output change) in a cycle (128 counts of the source clock).

PWMAD is a register used to set the additional pulse. When PWMAD is "1", an additional pulse that corresponds to 1 count of the source clock is added to the 2 × n-th duty pulse (n=1, 2, 3...). In other words, the 2 × n-th duty pulse has the output of PWMDUTY+1.

The additional pulse is not added when PWMAD is "0".

Downloaded from Arrow.com.

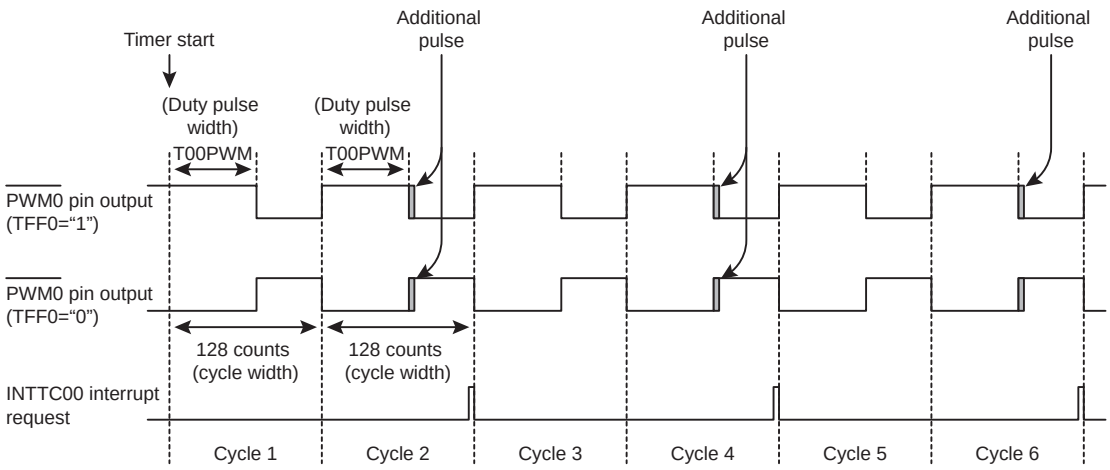


Figure 16-5 $\overline{\text{PWM0}}$ Pulse Output

Set the initial state of the $\overline{\text{PWM0}}$ pin at T00MOD<TFF0>. Setting T00MOD<TFF0> to "0" selects the "L" level as the initial state of the $\overline{\text{PWM0}}$ pin. Setting T00MOD<TFF0> to "1" selects the "H" level as the initial state of the $\overline{\text{PWM0}}$ pin. If the $\overline{\text{PWM0}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T00MOD<TFF0> is output to the $\overline{\text{PWM0}}$ pin. Table 16-6 shows the list of output levels of the $\overline{\text{PWM0}}$ pin.

Table 16-6 List of Output Levels of $\overline{\text{PWM0}}$ Pin

TFF0	$\overline{\text{PWM0}}$ pin output level			
	Before the start of operation (initial state)	T00PWM <PWMDUTY> matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

And by setting "1" to T001CR<OUTAND> bit, a logical product (AND) pulse of TC00 and TC01's output can be output to $\overline{\text{PWM0}}$ pin. By using this function, the remote-control waveform can be created eaily.

16.4.3.2 Operations

Setting T001CR<T00RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the lower 7 bits of the up counter value and the value set to T00PWM<PWMDUTY> is detected, the output of the $\overline{\text{PWM0}}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PWM0}}$ pin changes from the "L" to "H" level. When T00MOD<TFF0> is "1", the $\overline{\text{PWM0}}$ pin changes from the "H" to "L" level.

If T00PWM<PWMAD> is "1", an additional pulse that corresponds to 1 count of the source clock is added at the 2 × n-th match detection (n=1, 2, 3...). In other words, the $\overline{\text{PWM0}}$ pin output is reversed at the timing of T00PWM<PWMDUTY>+1. When T00MOD<TFF0> is "0", the period of the "L" level becomes longer than the value set to T00<PWMDUTY> by 1 source clock. When T00MOD<TFF0> is "1", the period of the "H" level becomes longer than the value set to T00PWM<PWMDUTY> by 1 source clock. This function allows two cycles of output pulses to be handled with a resolution nearly equivalent to 8 bits.

No additional pulse is inserted when T00PWM<PWMAD> is "0".

Subsequently, the up counter continues counting up. When the up counter value reaches 128, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of the $\overline{\text{PWM0}}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{\text{PWM0}}$ pin changes from the "H" to "L" level. When

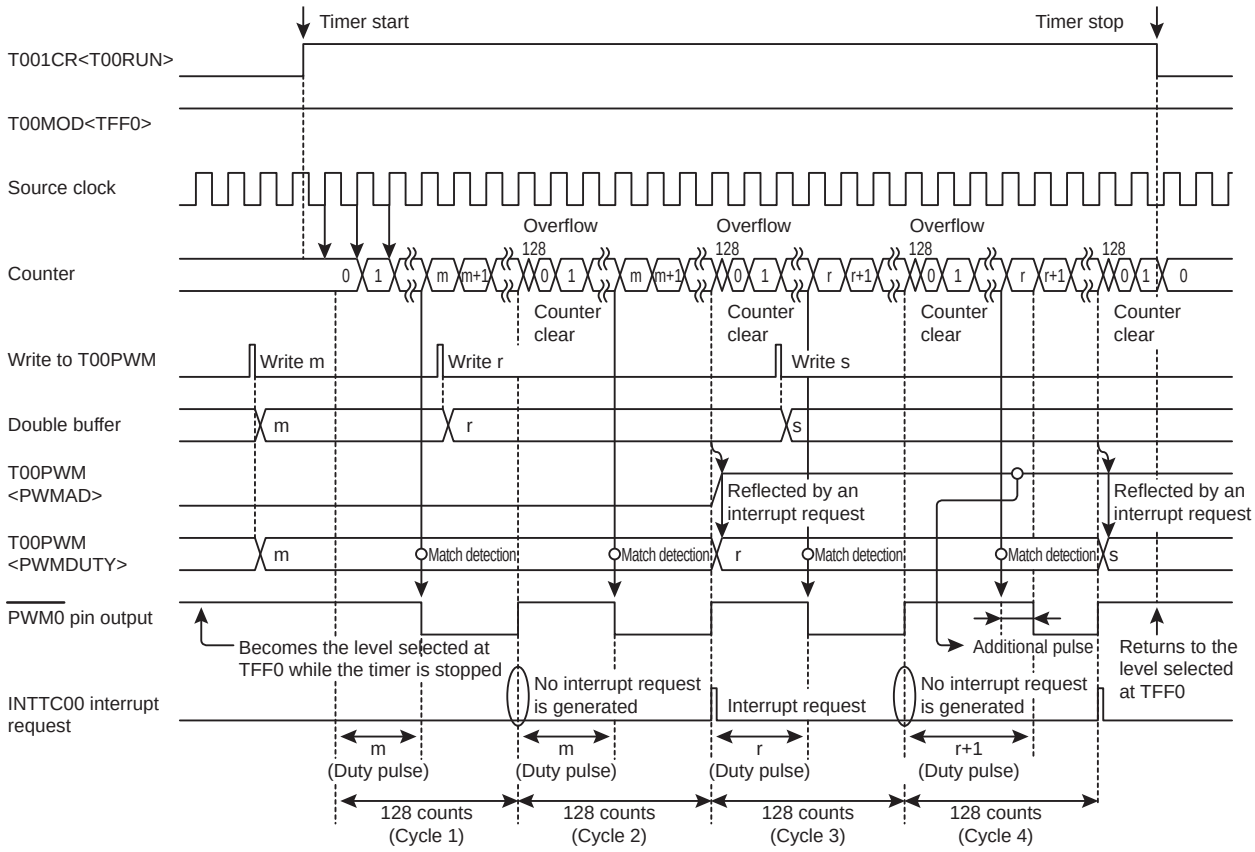
Downloaded from Arrow.com.

T00MOD<TFF0> is "1", the $\overline{\text{PWM0}}$ pin changes from the "L" to "H" level. If the 2 × n-th overflow occurs at this time, an INTTC00 interrupt request is generated. (No interrupt request is generated at the 2 × n-th -1 overflow.) Subsequently, the up counter continues counting up.

When T001CR<T00RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x00". The $\overline{\text{PWM0}}$ pin returns to the level selected at T00MOD<TFF0>.

(Example) Operate TC00 in the 8-bit PWM mode with the operation clock of fcgck/2 and output a duty pulse nearly equivalent to 11.6 μs (fcgck = 10 MHz)
(Actually, output a total duty pulse of 23.2 μs in 2 cycles (51.2 μs))

```
SET     (P9FC).4          ; Sets P9FC4 to "1"
SET     (P9CR).4          ; Sets P9CR4 to "1"
LD      (POFFCR0),0x10    ; Sets TC001EN to "1"
DI                        ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4          ; Sets the INTTC00 interrupt enable register to "1"
EI                        ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xF2     ; Selects the 8-bit PWM mode and fcgck/2
LD      (T00PWM),0x74     ; Sets the timer register (duty pulse)
                          ; (11.6μs × 2) / (2/fcgck) = 0x74
SET     (T001CR).0        ; Starts TC00
```



When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 16-6 8-bit PWM Mode Timing Chart

Downloaded from Arrow.com.

16.4.3.3 Double buffer

The double buffer can be used for T00PWM by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T00PWM during the timer operation, the set value is first stored in the double buffer, and T00PWM is not updated immediately. T00PWM compares the previous set value with the up counter value. When the 2 × n-th overflow occurs, an INTTC00 interrupt request is generated and the double buffer set value is stored in T00PWM. Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on T00PWM, the value in the double buffer (the last set value) is read out, not the T00PWM value (the currently effective value).

  When a write instruction is executed on T00PWM while the timer is stopped, the set value is immediately stored in both the double buffer and T00PWM.

- When the double buffer is disabled

  When a write instruction is executed on T00PWM during the timer operation, the set value is immediately stored in T00PWM. Subsequently, the match detection is executed using a new set value. If the value set to T00PWM is smaller than the up counter value, the $\overline{PWM0}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM. Therefore, the timing of changing the $\overline{PWM0}$ pin may not be an integral multiple of the source clock (Figure 16-7). Similarly, if T00PWM is set during the additional pulse output, the timing of changing the $\overline{PWM0}$ pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

  When a write instruction is executed on T00PWM while the timer is stopped, the set value is immediately stored in T00PWM.

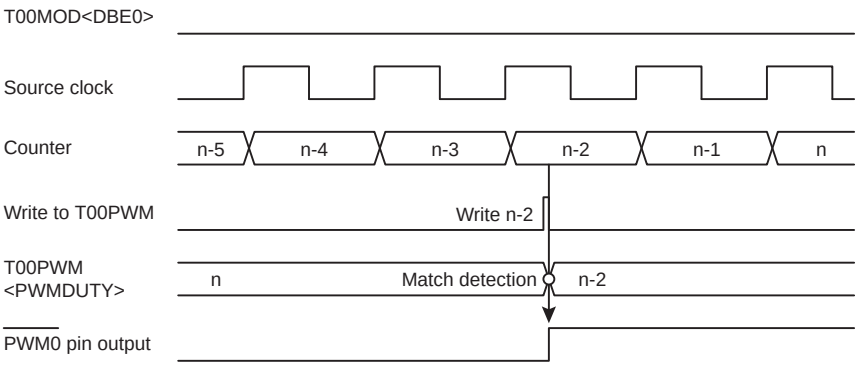


Figure 16-7 Operation When T00PWM and the Up Counter Have the Same Value

Downloaded from Arrow.com.

Table 16-7 Resolutions and Cycles in the 8-bit PWM Mode

T00MOD <TCK0>	Source clock [Hz]			Resolution		7-bit cycle (period × 2)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$	204.8μs	488.2μs	26.2ms (52.4ms)	62.5ms (125ms)
001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	13.1ms (26.2ms)	31.3ms (62.5ms)
010	$fcgck/2^8$	$fcgck/2^8$	-	25.6μs	-	3.3ms (6.6ms)	-
011	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	819.2μs (1638.4μs)	-
100	$fcgck/2^4$	$fcgck/2^4$	-	1.6μs	-	204.8μs (409.6μs)	-
101	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	51.2μs (102.4μs)	-
110	fcgck/2	fcgck/2	-	200ns	-	25.6μs (51.2μs)	-
111	fcgck	fcgck	$fs/2^2$	100ns	122.1μs	12.8μs (25.6μs)	15.6ms (31.3ms)

Downloaded from Arrow.com.

16.4.4 8-bit programmable pulse generate (PPG) output mode

In the 8-bit PPG mode, the pulses with arbitrary duty and cycle are output by using the T00REG and T00PWM registers.

By setting the T001CR<OUTAND> register, a pulse that is a logical ANDed product of the TC00 and TC01 outputs can be output to the TC01 pin. This function facilitates the generation of remote-controlled waveforms, for example.

The operation of TC00 is described below, and the same applies to the operation of TC01.

16.4.4.1 Setting

TC00 is put into the 8-bit PPG mode by setting T00MOD<TCM0> to "11" and T001CR<TCAS> to "0". Set T00MOD<EIN0> to "0" and select the clock at T00MOD<TCK0>. Set the duty pulse width at T00PWM and the cycle width at T00REG.

Set T00MOD<DBE0> to "1" to use the double buffer.

Setting T001CR<T00RUN> to "1" starts the operation. After the timer is started, writing to T00MOD becomes invalid. Be sure to complete the required mode settings before starting the timer.

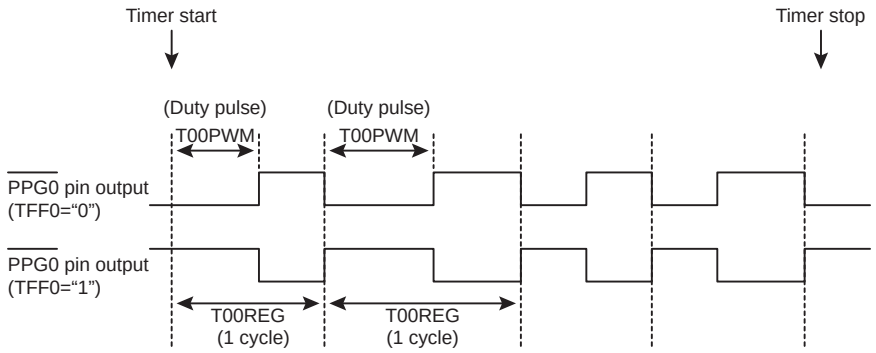


Figure 16-8 $\overline{\text{PPG0}}$ Pulse Output

Set the initial state of the $\overline{\text{PPG0}}$ pin at T00MOD<TFF0>. Setting T00MOD<TFF0> to "0" selects the "L" level as the initial state of the $\overline{\text{PPG0}}$ pin. Setting T00MOD<TFF0> to "1" selects the "H" level as the initial state of the $\overline{\text{PPG0}}$ pin. If the $\overline{\text{PPG0}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T00MOD<TFF0> is output to the $\overline{\text{PPG0}}$ pin. Table 16-8 shows the list of output levels of the $\overline{\text{PPG0}}$ pin.

Table 16-8 List of Output Levels of $\overline{\text{PPG0}}$ Pin

TFF0	$\overline{\text{PPG0}}$ pin output level			
	Before the start of operation (initial state)	T00PWM matched	T00REG matched	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

Setting the T001CR<OUTAND> bit to "1" allows the $\overline{\text{PPG0}}$ pin to output a pulse that is a logical ANDed product of the TC00 and TC01 outputs.

Downloaded from Arrow.com.

16.4.4.2 Operation

Setting T001CR<T00RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the internal up counter value and the value set to T00PWM is detected, the output of the $\overline{PPG0}$ pin is reversed. When T00MOD<TFF0> is "0", the $\overline{PPG0}$ pin changes from the "L" to "H" level. When T00MOD<TFF0> is "1", the $\overline{PPG0}$ pin changes from the "H" to "L" level.

Subsequently, the up counter continues counting up. When a match between the up counter value and T00REG is detected, the output of the $\overline{PPG0}$ pin is reversed again. When T00MOD<TFF0> is "0", the $\overline{PPG0}$ pin changes from the "H" to "L" level. When T00MOD<TFF0> is "1", the $\overline{PPG0}$ pin changes from the "L" to "H" level. At this time, an INTTC00 interrupt request is generated.

When T001CR<T00RUN> is set to "0" during the operation, the up counter is stopped and cleared to "0x00". The $\overline{PPG0}$ pin returns to the level selected at T00MOD<TFF0>.

16.4.4.3 Double buffer

The double buffer can be used for T00PWM and T00REG by setting T00MOD<DBE0>. The double buffer is disabled by setting T00MOD<DBE0> to "0" or enabled by setting T00MOD<DBE0> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T00PWM (T00REG) during the timer operation, the set value is first stored in the double buffer, and T00PWM (T00REG) is not updated immediately. T00PWM (T00REG) compares the previous set value with the up counter value. When an INTTC00 interrupt request is generated, the double buffer set value is stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on T00PWM (T00REG), the value in the double buffer (the last set value) is read out, not the T00PWM (T00REG) value (the currently effective value).

  When a write instruction is executed on T00PWM (T00REG) while the timer is stopped, the set value is immediately stored in both the double buffer and T00PWM (T00REG).

- When the double buffer is disabled

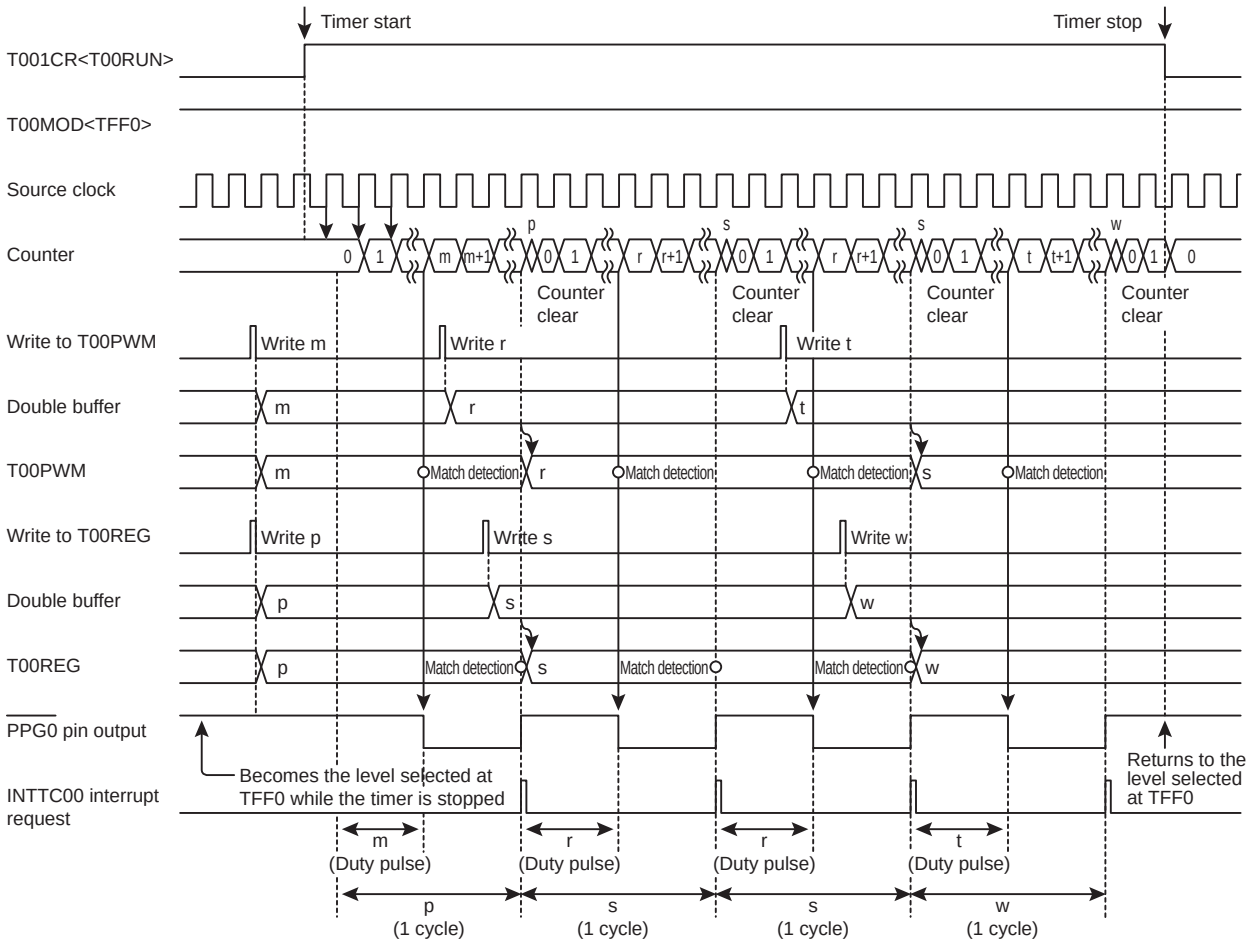
  When a write instruction is executed on T00PWM (T00REG) during the timer operation, the set value is immediately stored in T00PWM (T00REG). Subsequently, the match detection is executed using a new set value. If the value set to T00PWM (T00REG) is smaller than the up counter value, the $\overline{PPG0}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T00PWM (T00REG) is equal to the up counter value, the match detection is executed immediately after data is written into T00PWM (T00REG). Therefore, the timing of changing the $\overline{PPG0}$ pin may not be an integral multiple of the source clock (Figure 16-10). If these are problems, enable the double buffer.

  When a write instruction is executed on T00PWM (T00REG) while the timer is stopped, the set value is immediately stored in T00PWM (T00REG).

Downloaded from Arrow.com.

(Example) Operate TC00 in the 8-bit PPG mode with the operation clock of fcgck/2 and output the 8μs duty pulse in 32μs cycles (fcgck = 10 MHz)

```
SET     (P9FC).4          ; Sets P9FC4 to "1"
SET     (P9CR).4          ; Sets P9CR4 to "1"
LD      (POFFCR0),0x10    ; Sets TC001EN to "1"
DI                        ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4          ; Sets the INTTC00 interrupt enable register to "1"
EI                        ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xF3     ; Selects the 8-bit PPG mode and fcgck/2
LD      (T00REG),0xA0     ; Sets the timer register (cycle)
                          ; 32μs / (2/fcgck) = 0xA0
LD      (T00PWM),0x28     ; Sets the timer register (duty pulse)
                          ; 8μs / (2/fcgck) = 0x28
SET     (T001CR).0        ; Starts TC00
```



When the double buffer is enabled (T00MOD<DBE0>="1")

Figure 16-9 8-bit PPG Mode Timing Chart

Downloaded from Arrow.com.

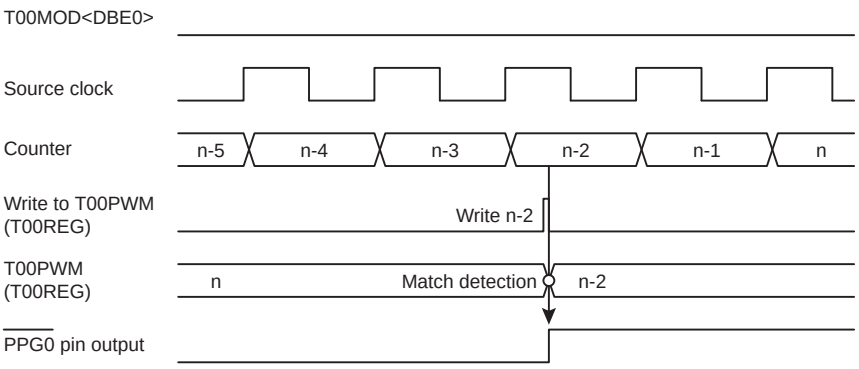


Figure 16-10 Operation When T00PWM (T00REG) and the Up Counter Have the Same Value

Downloaded from Arrow.com.

16.4.5 16-bit timer mode

In the 16-bit timer mode, TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer.

16.4.5.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

The 16-bit timer mode is activated by setting T01MOD<TCM1> to "00" or "01" and T01MOD<EIN1> to "0". Select the source clock at T01MOD<TCK1>.

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are reflected on the double buffer or T01+00REG when a write instruction is executed on T01REG. Be sure to execute the write instructions on T00REG and T01REG in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

16.4.5.2 Operations

Setting T001CR<T01RUN> to "1" allows the 16-bit up counter to increment based on the selected internal source clock. When a match between the up counter value and the T00+01REG set value is detected, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Setting T001CR<T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

16.4.5.3 Double buffer

The double buffer can be used for T01+00REG by setting T01MOD<DBE1>. The double buffer is disabled by setting T01MOD<DBE1> to "0" or enabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When write instructions are executed on T00REG and T01REG in this order during the timer operation, the set value is first stored in the double buffer, and T01+00REG is not updated immediately. T01+00REG compares the previous set value with the up counter value. When the values are matched, an INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00REG. Subsequently, the match detection is executed using a new set value.

  When write instructions are executed on T00REG and T01REG in this order while the timer is stopped, the set value is immediately stored in both the double buffer and T01+00REG.

- When the double buffer is disabled

  When write instructions are executed on T00REG and T01REG in this order during the timer operation, the set value is immediately stored in T01+00REG. Subsequently, the match detection is executed using a new set value.

  If the value set to T01+00REG is smaller than the up counter value, the match detection is executed using a new set value after the up counter overflows. Therefore, the interrupt request inter-

Downloaded from Arrow.com.

val may be longer than the selected time. If the value set to T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00REG. Therefore, the interrupt request interval may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

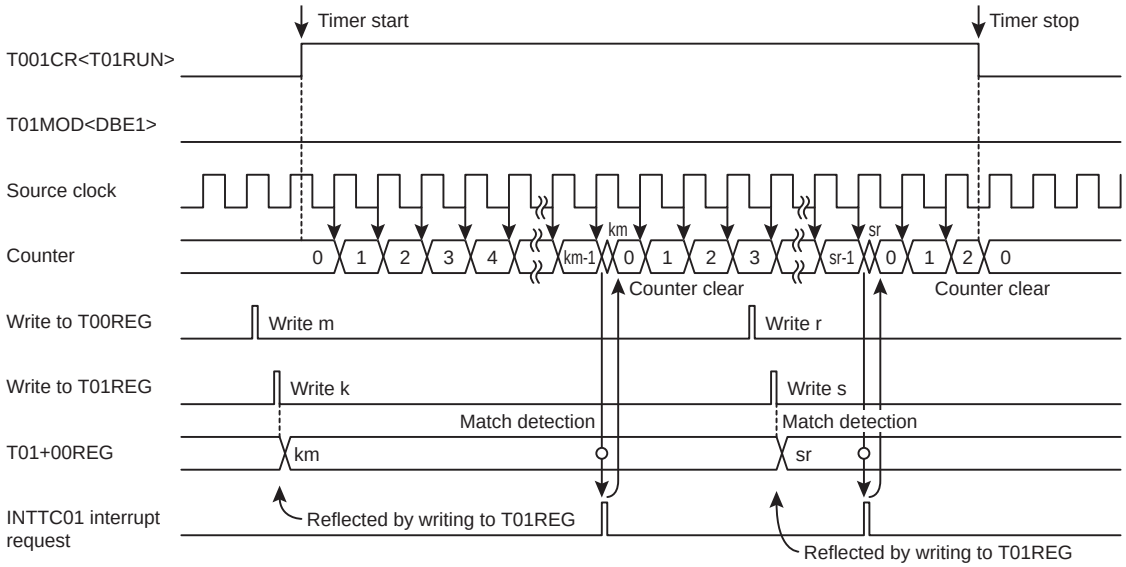
When write instructions are executed on T00REG and T01REG in this order while the timer is stopped, the set value is immediately stored in T01+00REG.

When a read instruction is executed on T01+00REG, the last value written into T01+00REG is read out, regardless of the T00MOD<DBE1> setting.

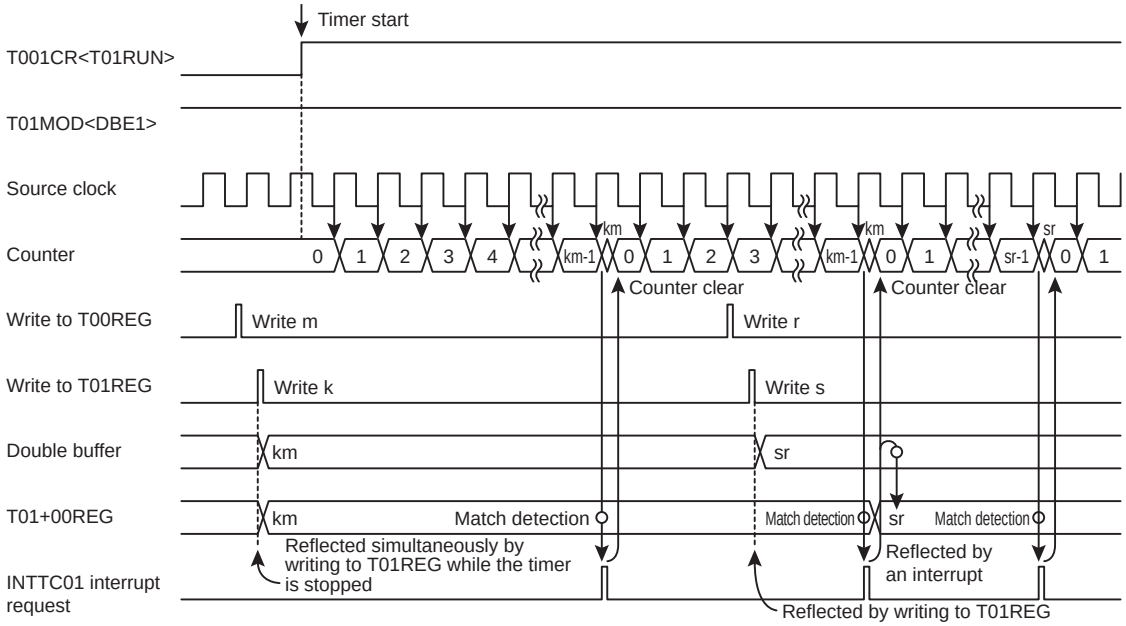
(Example) Operate TC00 and TC01 in the 16-bit timer mode with the operation clock of fcgck/2 [Hz] and generate interrupts at 96 μs intervals (fcgck = 10 MHz)

```
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF0       ; Selects the 16-bit timer mode and fcgck/2
LD      (T00REG),0xE0       ; Sets the timer register (96μs / (2/fcgck) = 0x1E0)
LD      (T01REG),0x01       ; Sets the timer register
LD      (T001CR),0x06       ; Starts TC00 and TC001 (16-bit mode)
```


2026-04-21
Downloaded from Arrow.com.



When the double buffer is disabled (T01MOD<DBE1>="0")



When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 16-11 16-bit Timer Counter Timing Chart

Downloaded from Arrow.com.

Table 16-9 16-bit Timer Mode Resolution and Maximum Time Setting

T01MOD <TCK1>	Source clock [Hz]			Resolution		Maximum time setting	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$	204.8μs	488.2μs	13.4s	32s
001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	6.7s	16s
010	$fcgck/2^8$	$fcgck/2^8$	-	25.6μs	-	1.7s	-
011	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	419.4ms	-
100	$fcgck/2^4$	$fcgck/2^4$	-	1.6μs	-	104.9ms	-
101	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	26.2ms	-
110	fcgck/2	fcgck/2	-	200ns	-	13.1ms	-
111	fcgck	fcgck	$fs/2^2$	100ns	122.1μs	6.6ms	8s

Downloaded from Arrow.com.

16.4.6 16-bit event counter mode

In the 16-bit event counter mode, the up counter counts up at the falling edge of the input to the TC00 pin. TC00 and TC01 are cascaded to form a 16-bit timer counter, which can measure a longer period than an 8-bit timer.

16.4.6.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode.

The 16-bit timer mode is activated by setting T01MOD<TCM1> to "00" or "01" and T01MOD<EIN0> to "1".

Set the count value to be used for the match detection as a 16-bit value at the timer registers T00REG and T01REG. Set the lower 8 bits of the 16-bit value at T00REG and set the higher 8 bits at T01REG. (Hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG.) The timer register settings are reflected on the double buffer or T01+00REG when a write instruction is executed on T01REG. Be sure to execute the write instructions on T00REG and T01REG in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

16.4.6.2 Operations

Setting T001CR<T01RUN> to "1" allows the 16-bit up counter to increment at the falling edge of the TC00 pin. When a match between the up counter value and the T00+01REG set value is detected, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000". After being cleared, the up counter restarts counting. Setting T001CR<T01RUN> to "0" during the timer operation makes the up counter stop counting and be cleared to "0x0000".

The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or $fs/2^4$ [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

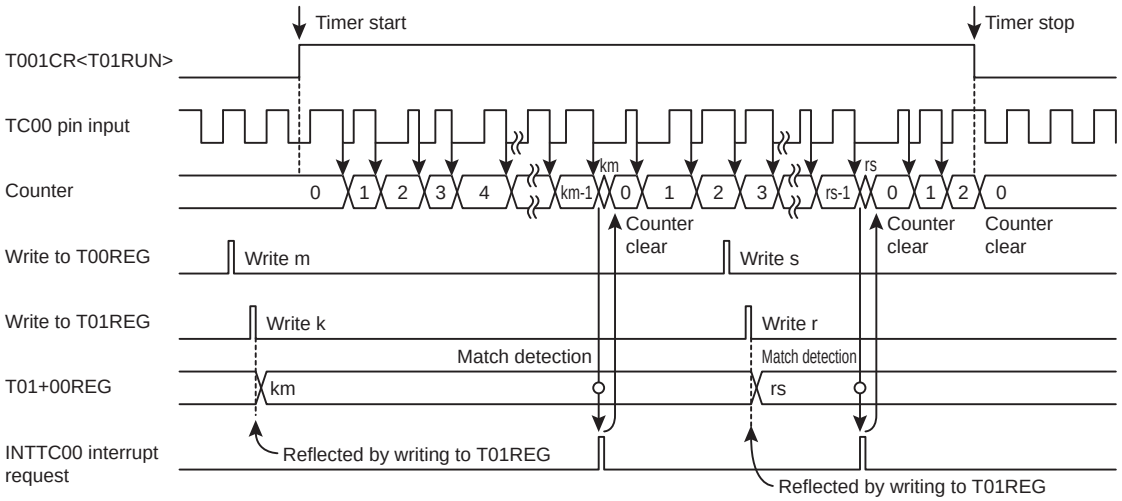
16.4.6.3 Double buffer

Refer to 16.4.5.3.

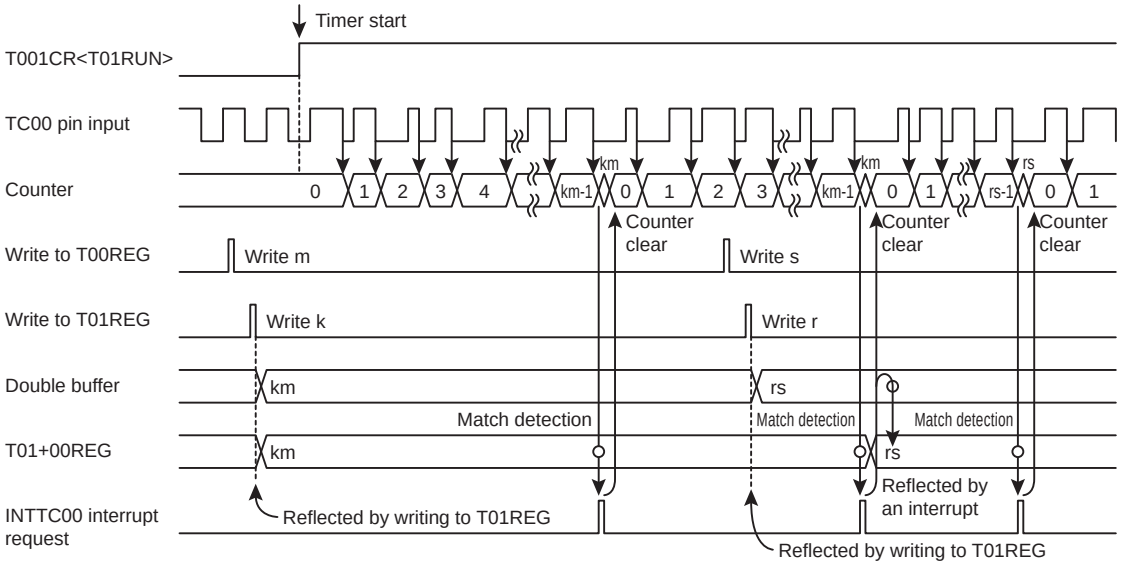
Downloaded from Arrow.com.

(Example) Operate TC00 and TC01 in the 16-bit event counter mode and generate an interrupt each time the 384th falling edge is detected at the TC00 pin

```
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T00MOD),0xC4       ; Selects the 16-bit event counter mode
LD      (T00REG),0x80       ; Sets the timer register
LD      (T01REG),0x10       ; Sets the timer register
LD      (T001CR),0x06       ; Starts TC00 and TC001 (16-bit mode)
```



When the double buffer is disabled (T01MOD<DBE1>="0")



When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 16-12 16-bit Event Counter Mode Timing Chart

Downloaded from Arrow.com.

16.4.7 12-bit pulse width modulation (PWM) output mode

In the 12-bit PWM output mode, TC00 and TC01 are cascaded to output the pulse-width modulated pulses with a resolution of 8 bits. An additional pulse of 4 bits can be inserted, which enables PWM output with a resolution nearly equivalent to 12 bits.

16.4.7.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit timer mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit timer mode.

The 12-bit PWM mode is selected by setting T01MOD<TCM1> to "10". To use the internal clock as the source clock, set T01MOD<EIN1> to "0" and select the clock at T01MOD<TCK1>. To use an external clock as the source clock, set T01MOD<EIN1> to "1".

Set T01MOD<DBE1> to "1" to use the double buffer.

Setting T001CR<T01RUN> to "1" starts the operation. After the timer is started, writing to T01MOD becomes invalid. Be sure to complete the required mode settings before starting the timer. (Make settings when T001CR<T00RUN> and <T01RUN> are "0".)

Set the count value to be used for the match detection and the additional pulse value as a 12-bit value at the timer registers T00PWM and T01PWM. Set bits 11 to 8 of the 12-bit value at the lower 4 bits of T01PWM and set bits 7 to 0 at T00PWM. Refer to the following table for the register configuration. Hereinafter, the 12-bit value specified by the combined setting of T00PWM and T01PWM is indicated as T01+00PWM. The timer register settings are reflected on the double buffer or T01+00PWM when a write instruction is executed on T01PWM. Be sure to execute the write instructions on T00PWM and T01PWM in this order. (When data is written to the high-order register, the set values of the low-order and high-order registers become effective at the same time.)

Timer register 00

T00PWM		7	6	5	4	3	2	1	0
(0x00028)	Bit Symbol	PWMDUTYL				PWMAD3	PWMAD2	PWMAD1	PWMAD0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

Timer register 01

T01PWM		7	6	5	4	3	2	1	0
(0x00029)	Bit Symbol					PWMDUTYH			
	Read/Write					R/W	R/W	R/W	R/W
	After reset	1	1	1	1	1	1	1	1

Bits 7 to 4 of T01PWM are not used in the 12-bit PWM mode. However, data can be written to these bits of T01PWM and the written values are read out as they are when the bits are read. Normally, set these bits to "0".

PWMDUTYH and PWMDUTYL are 4-bit registers. They are combined to set an 8-bit value of duty pulse width (time before the first change in the output) for one cycle (256 counts of the source clock). Hereinafter, an 8-bit value specified by the combined setting of PWMDUTYH and PWMDUTYL is indicated as PWMDUTY.

PWMAD3 to 0 are the additional pulse setting register. Additional pulses can be inserted in specific cycles of the duty pulse by setting each bit to "1". The additional pulses are inserted in the positions listed

Downloaded from Arrow.com.

in Table 16-10. PWMAD 3 to 0 can be combined to specify the number of times of inserting the additional pulses in 16 cycles to any number from 1 to 16. Examples of inserting additional pulses are shown in Figure 16-13.

Table 16-10 Cycles in Which Additional Pulses Are Inserted

	Cycles in which additional pulses are inserted among cycles 1 to 16
PWMAD0="1"	9
PWMAD1="1"	5, 13
PWMAD2="1"	3, 7, 11, 15
PWMAD3="1"	2, 4, 6, 8, 10, 12, 14, 16

Set the initial state of the $\overline{\text{PWM1}}$ pin at T01MOD<TFF1>. Setting T01MOD<TFF1> to "0" selects the "L" level as the initial state of the $\overline{\text{PWM1}}$ pin. Setting T01MOD<TFF1> to "1" selects the "H" level as the initial state of the $\overline{\text{PWM1}}$ pin. If the $\overline{\text{PWM1}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD<TFF1> is output to the PWM1 pin. Table 16-11 shows the list of output levels of the $\overline{\text{PWM1}}$ pin.

Table 16-11 List of Output Levels of $\overline{\text{PWM1}}$ Pin

TFF1	$\overline{\text{PWM1}}$pin output level			
	Before the start of operation (initial state)	PWMDUTY matched (after the additional pulse)	Overflow	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

Downloaded from Arrow.com.

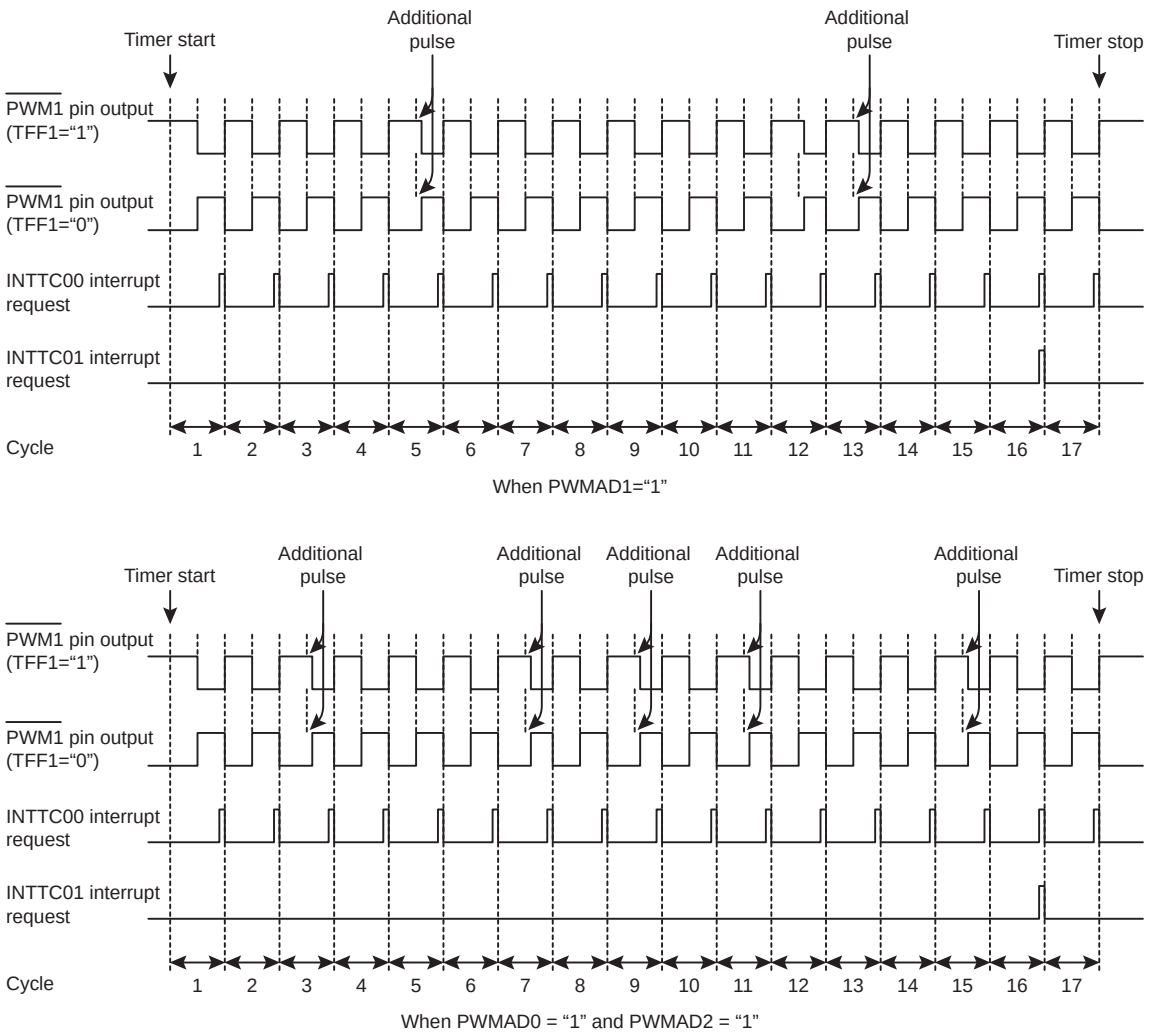


Figure 16-13 Examples of Inserting Additional Pulses

16.4.7.2 Operations

Setting T001CR<T01RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the lower 8 bits of the up counter value and the value set to PWMDUTY is detected, the output of the $\overline{\text{PWM1}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PWM1}}$ pin changes from the "L" to "H" level. When T01MOD<TFF1> is "1", the $\overline{\text{PWM1}}$ pin changes from the "H" to "L" level.

If any of PWMAD3 to 0 is "1", an additional pulse that corresponds to 1 count of the source clock is inserted in specific cycles of the duty pulse. In other words, the $\overline{\text{PWM1}}$ pin output is reversed at the timing of PWMDUTY+1. When T00MOD<TFF0> is "0", the period of the "L" level becomes longer than the value set to PWMDUTY by 1 source clock. When T00MOD<TFF0> is "1", the period of the "H" level becomes longer than the value set to PWMDUTY by 1 source clock. This function allows 16 cycles of output pulses to be handled with a resolution nearly equivalent to 12 bits.

No additional pulse is inserted when PWMAD3 to 0 are all "0".

Subsequently, the up counter continues counting up. When the up counter value reaches 256, an overflow occurs and the up counter is cleared to "0x00". At the same time, the output of the $\overline{\text{PWM1}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PWM1}}$ pin changes from the "H" to "L" level. When T01MOD<TFF1> is "1", the $\overline{\text{PWM1}}$ pin changes from the "L" to "H" level. At this time, an INTTC00 interrupt request is generated (an INTTC00 interrupt request is generated each time an overflow occurs.) An INTTC01 interrupt request is generated at the 16 × n-th overflow (n=1, 2, 3...). Subsequently, the up counter continues counting up.

Downloaded from Arrow.com.

When T001CR<T01RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x00". The $\overline{\text{PWM1}}$ pin returns to the level selected at T01MOD<TFF1>.

When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or $fs/2^4$ [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

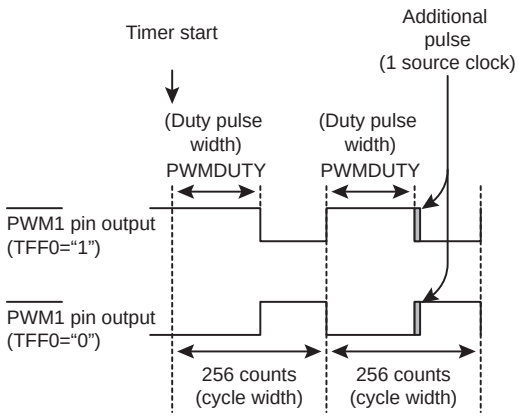


Figure 16-14 $\overline{\text{PWM1}}$ Pin Output

16.4.7.3 Double buffer

The double buffer can be used for T01+00PWM by setting T01MOD<DBE1>. The double buffer is disabled by setting T01MOD<DBE1> to "0" or enabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When write instructions are executed on T00PWM and T01PWM in this order during the timer operation, the set value is first stored in the double buffer, and T01+00PWM is not updated immediately. T01+00PWM compares the previous set value with the up counter value. When the 16 × n-th overflow occurs, an INTTC01 interrupt request is generated and the double buffer set value is stored in T01+00PWM. Subsequently, the match detection is executed using a new set value.

  When a read instruction is executed on T01+00PWM (T00REG), the value in the double buffer (the last set value) is read out, not the T01+00PWM value (the currently effective value).

  When write instructions are executed on T00PWM and T01PWM in this order while the timer is stopped, the set value is immediately stored in both the double buffer and T01+00PWM.

- When the double buffer is disabled

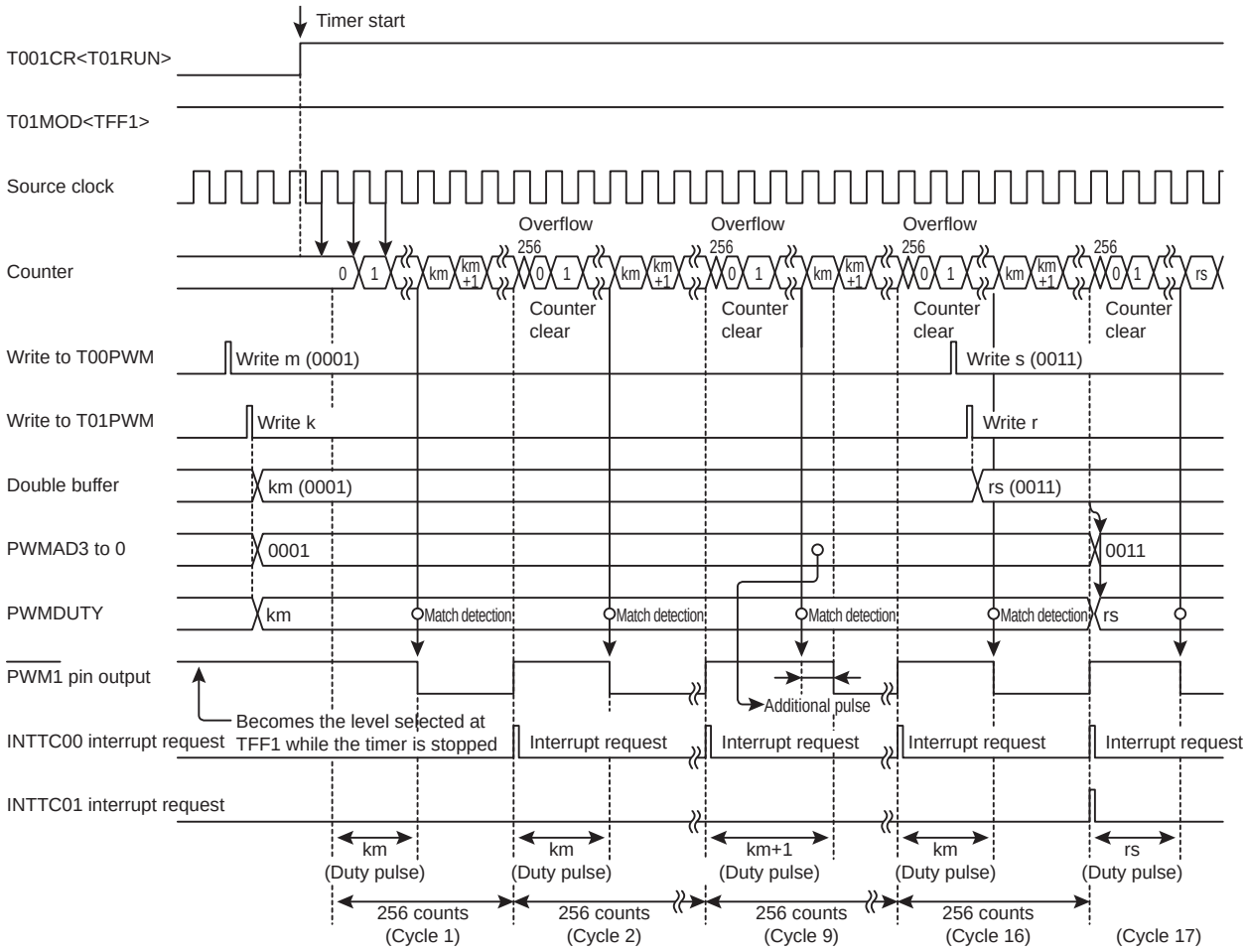
  When write instructions are executed on T00PWM and T01PWM in this order during the timer operation, the set value is immediately stored in T01+00PWM. Subsequently, the match detection is executed using a new set value. If the value set to T01+00PWM is smaller than the up counter value, the $\overline{\text{PWM1}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T01+00PWM is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM. Therefore, the timing of changing the $\overline{\text{PWM1}}$ pin may not be an integral multiple of the source clock. Similarly, if T01+00PWM is set during the additional pulse output, the timing of changing the $\overline{\text{PWM1}}$ pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

  When write instructions are executed on T00PWM and T01PWM in this order while the timer is stopped, the set value is immediately stored in T01+00PWM.

Downloaded from Arrow.com.

(Example) Operate TC00 and TC01 in the 12-bit PWM mode with the operation clock of fcgck/2 and output a duty pulse nearly equivalent to 14.0625 μs in 51.2μs cycles (fcgck = 10 MHz)
(Actually, output a duty pulse of 225 μs in total in 16 cycles (819.2 μs))

```
SET     (P9FC).5          ; Sets P9FC5 to "1"
SET     (P9CR).5          ; Sets P9CR5 to "1"
LD      (POFFCR0),0x10    ; Sets TC001EN to "1"
DI                        ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4          ; Sets the INTTC00 interrupt enable register to "1"
EI                        ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF2     ; Selects the 12-bit PWM mode and fcgck/2
LD      (T00PWM),0x65     ; Sets the timer register (duty pulse)
                          ; (14.0625μs × 16) / (2/fcgck) = 0x465
LD      (T01PWM),0x04     ; Sets the timer register (duty pulse)
LD      (T001CR),0x06     ; Starts TC00 and TC01
```



When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 16-15 12-bit PWM Mode Timing Chart

Downloaded from Arrow.com.

Table 16-12 Resolutions and Cycles in the 12-bit PWM Mode

T01MOD <TCK1>	Source clock [Hz]			Resolution		8-bit cycle (period × 16)	
	NORMAL1/2 or IDLE1/2 mode		SLOW1/2 or SLEEP1 mode	fcgck=10MHz	fs=32.768kHz	fcgck=10MHz	fs=32.768kHz
	SYSCR1<DV9CK> = "0"	SYSCR1<DV9CK> = "1"					
000	$fcgck/2^{11}$	$fs/2^4$	$fs/2^4$	204.8μs	488.2μs	52.4ms (838.9ms)	125ms (2000ms)
001	$fcgck/2^{10}$	$fs/2^3$	$fs/2^3$	102.4μs	244.1μs	26.2ms (419.4ms)	62.5ms (1000ms)
010	$fcgck/2^8$	$fcgck/2^8$	-	25.6μs	-	6.6ms (104.9ms)	-
011	$fcgck/2^6$	$fcgck/2^6$	-	6.4μs	-	1.6ms (26.2ms)	-
100	$fcgck/2^4$	$fcgck/2^4$	-	1.6μs	-	409.6μs (6.6ms)	-
101	$fcgck/2^2$	$fcgck/2^2$	-	400ns	-	102.4μs (1.6ms)	-
110	fcgck/2	fcgck/2	-	200ns	-	51.2μs (819.2μs)	-
111	fcgck	fcgck	$fs/2^2$	100ns	122.1μs	25.6μs (409.6μs)	31.3ms (500ms)

Downloaded from Arrow.com.

16.4.8 16-bit programmable pulse generate (PPG) output mode

In the 16-bit PPG mode, TC00 and TC01 are cascaded to output the pulses that have a resolution of 16 bits and arbitrary pulse width and duty. Two 16-bit registers, T01+00REG and T01+00PWM, are used to output the pulses. This enables output of longer pulses than an 8-bit timer.

16.4.8.1 Setting

Setting T001CR<TCAS> to "1" connects TC00 and TC01 and activates the 16-bit mode. All the settings of TC00 are ignored and those of TC01 are effective in the 16-bit mode.

The 16-bit PPG mode is selected by setting T01MOD<TCM1> to "11". To use the internal clock as the source clock, set T01MOD<EIN1> to "0" and select the clock at T01MOD<TCK1>. To use an external clock as the source clock, set T01MOD<EIN0> to "1".

Set T01MOD<DBE1> to "1" to use the double buffer.

Set the count value that corresponds to a cycle as a 16-bit value at the timer registers T01REG and T00REG. Set the count value that corresponds to a duty pulse as a 16-bit value at T01PWM and T00PWM (hereinafter, the 16-bit value specified by the combined setting of T01REG and T00REG is indicated as T01+00REG, and the 16-bit value specified by the combined setting of T01PWM and T00PWM is indicated as T01+00PWM). The timer register settings are reflected on the double buffer or T01+00PWM and T01+00REG when a write instruction is executed on T01PWM. Be sure to execute the write instructions on T00REG, T01REG and T00PWM before executing a write instruction on T01PWM. (When data is written to T01PWM, the set values of the four timer registers become effective at the same time.)

Set the initial state of the $\overline{\text{PPG1}}$ pin at T01MOD<TFF1>. Setting T01MOD<TFF1> to "0" selects the "L" level as the initial state of the $\overline{\text{PPG1}}$ pin. Setting T01MOD<TFF1> to "1" selects the "H" level as the initial state of the $\overline{\text{PPG1}}$ pin. If the $\overline{\text{PPG1}}$ pin is set as the function output pin in the port setting while the timer is stopped, the value of T01MOD<TFF1> is output to the $\overline{\text{PPG1}}$ pin. Table 16-13 shows the list of output levels of the $\overline{\text{PPG1}}$ pin.

Table 16-13 List of Output Levels of $\overline{\text{PPG1}}$ Pin

TFF1	$\overline{\text{PPG1}}$ pin output level			
	Before the start of operation (initial state)	T01+00PWM matched	T01+00REG matched	Operation stopped (initial state)
0	L	H	L	L
1	H	L	H	H

16.4.8.2 Operations

Setting T001CR<T01RUN> to "1" allows the up counter to increment based on the selected source clock. When a match between the up counter value and the value set to T01+00PWM is detected, the output of the $\overline{\text{PPG1}}$ pin is reversed. When T01MOD<TFF1> is "0", the $\overline{\text{PPG1}}$ pin changes from the "L" to "H" level. When T01MOD<TFF1> is "1", the $\overline{\text{PPG1}}$ pin changes from the "H" to "L" level. At this time, an INTTC00 interrupt request is generated.

The up counter continues counting up. When a match between the up counter value and the value set to T01+00REG is detected, the output of the $\overline{\text{PPG1}}$ pin is reversed again. When T01MOD<TFF1> is "0", the $\overline{\text{PPG1}}$ pin changes from the "H" to "L" level. When T01MOD<TFF1> is "1", the $\overline{\text{PPG1}}$ pin changes from the "L" to "H" level. At this time, an INTTC01 interrupt request is generated and the up counter is cleared to "0x0000".

Downloaded from Arrow.com.

When T001CR<T01RUN> is set to "0" during the timer operation, the up counter is stopped and cleared to "0x0000". The $\overline{\text{PPG1}}$ pin returns to the level selected at T01MOD<TFF1>.

When an external source clock is selected, input the clock at the TC00 pin. The maximum frequency to be supplied is fcgck/2 [Hz] (in NORMAL1/2 or IDLE1/2 mode) or $fs/2^4$ [Hz] (in SLOW1/2 or SLEEP1 mode), and a pulse width of two machine cycles or more is required at both the "H" and "L" levels.

16.4.8.3 Double buffer

The double buffer can be used for T01+00PWM and T01+00REG by setting T01MOD<DBE1>. The double buffer is enabled by setting T01MOD<DBE1> to "0" or disabled by setting T01MOD<DBE1> to "1".

- When the double buffer is enabled

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM during the timer operation, the set values are first stored in the double buffer, and T01+00PWM and T01+00REG are not updated immediately. T01+00PWM and T01+00REG compare the previous set values with the up counter value. When a match between the up counter value and the T01+00REG set value is detected, an INTTC01 interrupt request is generated and the double buffer set values are stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new set values.

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM while the timer is stopped, the set values are immediately stored in both the double buffer and T01+00PWM and T01+00REG.

- When the double buffer is disabled

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM during the timer operation, the set values are immediately stored in T01+00PWM and T01+00REG. Subsequently, the match detection is executed using new set values.

  If the value set to T01+00PWM or T01+00REG is smaller than the up counter value, the $\overline{\text{PPG1}}$ pin is not reversed until the up counter overflows and a match detection is executed using a new set value. If the value set to T01+00PWM or T01+00REG is equal to the up counter value, the match detection is executed immediately after data is written into T01+00PWM and T01+00REG. Therefore, the timing of changing the $\overline{\text{PPG1}}$ pin may not be an integral multiple of the source clock. If these are problems, enable the double buffer.

  When a write instruction is executed on T01PWM after write instructions are executed on T00REG, T01REG and T00PWM while the timer is stopped, the set values are immediately stored in T01+00PWM and T01+00REG.

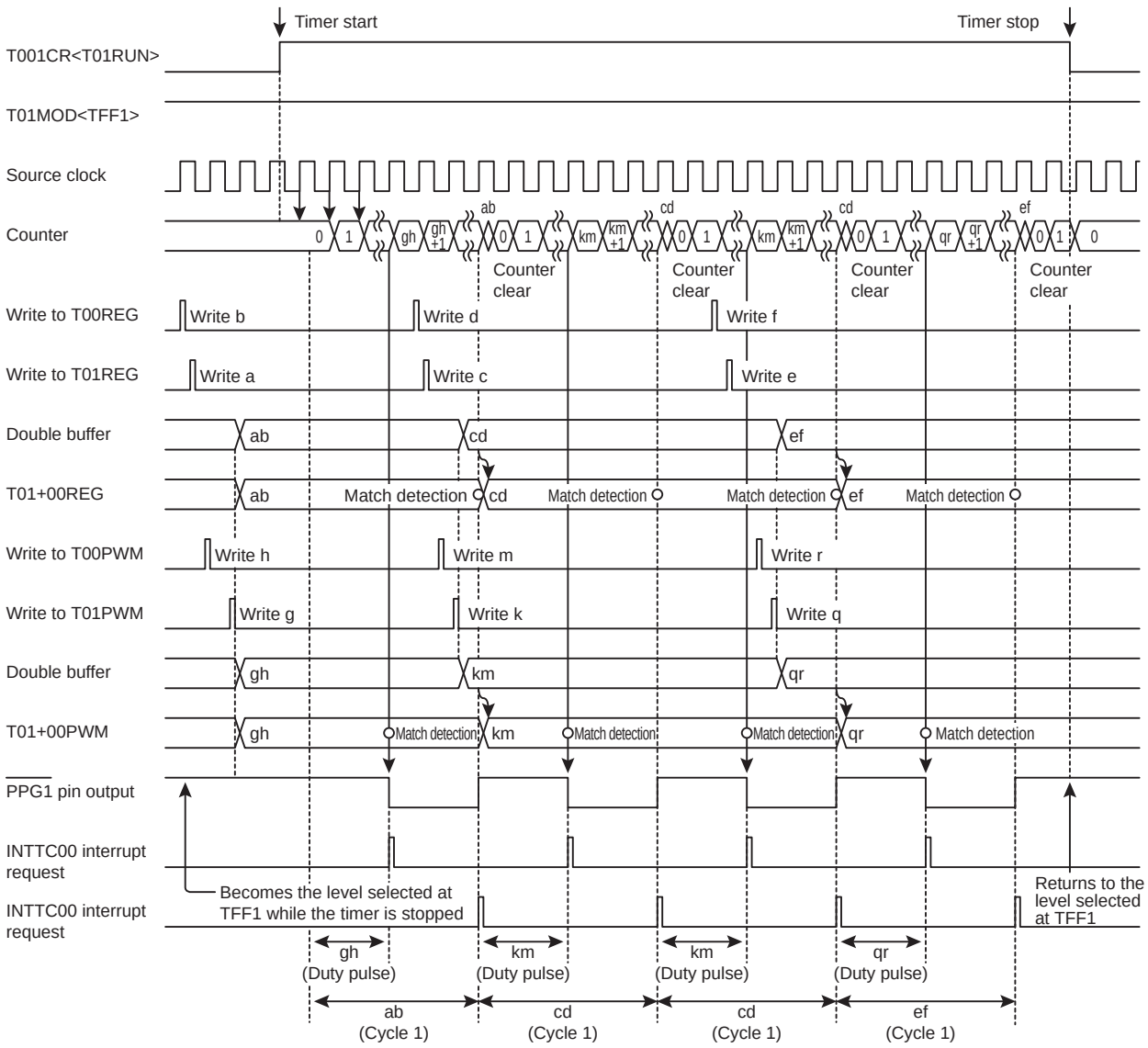
When read instructions are executed on T01+00PWM and T01+00REG, the last value written into T01+00REG is read out, regardless of the T00MOD<DBE1> setting.

Downloaded from Arrow.com.

(Example) Operate TC00 and TC01 in the 16-bit PPG mode with the operation clock of fcgck/2 and output the 68μs duty pulse in 96μs cycles (fcgck = 10 MHz)

```
SET     (P9FC).5            ; Sets P9FC5 to "1"
SET     (P9CR).5            ; Sets P9CR5 to "1"
LD      (POFFCR0),0x10      ; Sets TC001EN to "1"
DI                          ; Sets the interrupt master enable flag to "disable"
SET     (EIRH).4            ; Sets the INTTC00 interrupt enable register to "1"
EI                          ; Sets the interrupt master enable flag to "enable"
LD      (T01MOD),0xF3       ; Selects the 8-bit PPG mode and fcgck/2
LD      (T00REG),0xE0       ; Sets the timer register (cycle)
LD      (T01REG),0x01       ; Sets the timer register (cycle)
                            ; 96μs / (2/fcgck) = 0x01E0
LD      (T00PWM),0x54       ; Sets the timer register (duty pulse)
LD      (T01PWM),0x01       ; Sets the timer register (duty pulse)
                            ; 68μs / (2/fcgck) = 0x0154
LD      (T001CR),0x06       ; Starts TC00 and TC01
```


Downloaded from Arrow.com.



When the double buffer is enabled (T01MOD<DBE1>="1")

Figure 16-16 16-bit PPG Output Mode Timing Chart

Downloaded from Arrow.com.

17. Real Time Clock (RTC)

The real time clock is a function that generates interrupt requests at certain intervals using the low-frequency clock.

The number of interrupts is counted by the software to realize the clock function.

The real time clock can be used only in the operation modes where the low-frequency clock oscillates, except for SLEEP0.

17.1 Configuration

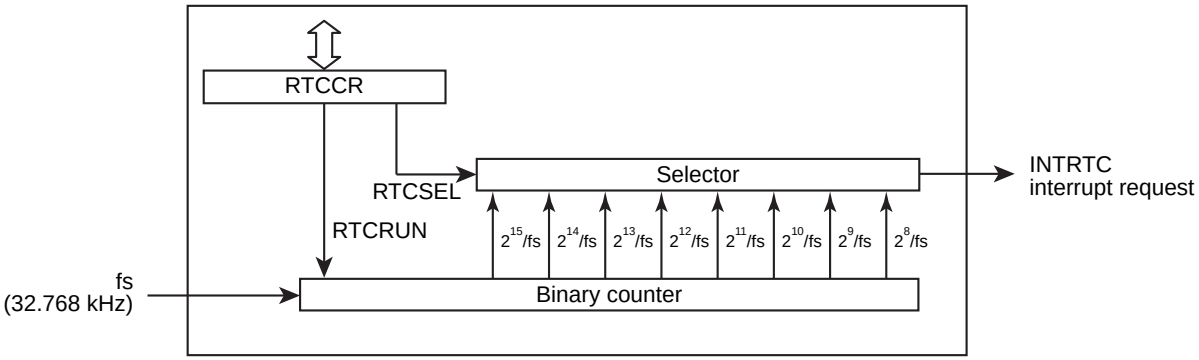


Figure 17-1 Real Time Clock

17.2 Control

The real time clock is controlled by following resisters.

Low power consumption register 2

POFFCR2		7	6	5	4	3	2	1	0
(0x00F76)	Bit Symbol	LCDEN	-	RTCEN	-	-	-	SIO1EN	SIO0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

LCDEN	LCD Control	0 1	Disable Enable
RTCEN	RTC Control	0 1	Disable Enable
SIO1EN	SIO1 Control	0 1	Disable Enable
SIO0EN	SIO0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

Real time clock control register

RTCCR (0x00FC8)

	7	6	5	4	3	2	1	0
Bit Symbol	-	-	-	-	RTCSEL			RTCRUN
Read/Write	R	R	R	R	R/W			R/W
After reset	0	0	0	0	0	0	0	0

RTCSEL	Selects the interrupt generation interval	000 : 2^{15}/fs (1.000 [s] @fs=32.768kHz) 001 : 2^{14}/fs (0.500 [s] @fs=32.768kHz) 010 : 2^{13}/fs (0.250 [s] @fs=32.768kHz) 011 : 2^{12}/fs (125.0 [ms] @fs=32.768kHz) 100 : 2^{11}/fs (62.50 [ms] @fs=32.768kHz) 101 : 2^{10}/fs (31.25 [ms] @fs=32.768kHz) 110 : 2^{9}/fs (15.62 [ms] @fs=32.768kHz) 111 : 2^{8}/fs (7.81 [ms] @fs=32.768kHz)
RTCRUN	Enables/disables the real time clock operation	0 : Disable 1 : Enable

Note 1: fs: Low-frequency clock [Hz]

Note 2: RTCCR<RTCSEL> can be rewritten only when RTCCR<RTCRUN> is "0". If data is written into RTCCR<RTCSEL> when RTCCR<RTCRUN> is "1", the existing data remains effective. RTCCR<RTCSEL> can be rewritten at the same time as enabling the real time clock, but it cannot be rewritten at the same time as disabling the real time clock.

Note 3: If the real time clock is enabled and when 1) SYSCR2<XTEN> is cleared to "0" to stop the low-frequency clock oscillation circuit or 2) the operation is changed to the STOP mode or the SLEEP0 mode, the data in RTCCR<RTCSEL> is maintained and RTCCR<RTCRUN> is cleared to "0".

17.3 Function

17.3.1 Low Power Consumption Function

Real time clock has the low power consumption registers (POFFCR2) that save power when the real time clock is not being used.

Setting POFFCR2<RTCEN> to "0" disables the basic clock supply to real time clock to save power. Note that this renders the real time clock unusable. Setting POFFCR2<RTCEN> to "1" enables the basic clock supply to real time clock and allows the real time clock to operate.

After reset, POFFCR2<RTCEN> are initialized to "0", and this renders the real time clock unusable. When using the real time clock for the first time, be sure to set POFFCR2<RTCEN> to "1" in the initial setting of the program (before the real time clock control registers are operated).

Do not change POFFCR2<RTCEN> to "0" during the real time clock operation. Otherwise real time clock may operate unexpectedly.

17.3.2 Enabling/disabling the real time clock operation

Setting RTCCR<RTCRUN> to "1" enables the real time clock operation. Setting RTCCR<RTCRUN> to "0" disables the real time clock operation.

RTCCR<RTCRUN> is cleared to "0" just after reset release.

17.3.3 Selecting the interrupt generation interval

The interrupt generation interval can be selected at RTCCR<RTCSEL>.

Downloaded from Arrow.com.

RTCCR<RTCSEL> can be rewritten only when RTCCR<RTCRUN> is "0". If data is written into RTCCR<RTCSEL> when RTCCR<RTCRUN> is "1", the existing data remains effective.

RTCCR<RTCSEL> can be rewritten at the same time as enabling the real time clock operation, but it cannot be rewritten at the same time as disabling the real time clock operation.

17.4 Real Time Clock Operation

17.4.1 Enabling the real time clock operation

Set the interrupt generation interval to RTCCR<RTCSEL>, and at the same time, set RTCCR<RTCRUN> to "1".

When RTCCR<RTCRUN> is set to "1", the binary counter for the real time clock starts counting of the low-frequency clock.

When the interrupt generation interval selected at RTCCR<RTCSEL> is reached, a real time clock interrupt request (INTRTC) is generated and the counter continues counting.

17.4.2 Disabling the real time clock operation

Clear RTCCR<RTCRUN> to "0".

When RTCCR<RTCRUN> is cleared to "0", the binary counter for the real time clock is cleared to "0" and stops counting of the low-frequency clock.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

18. Asynchronous Serial Interface (UART)

The TMP89FW24A contains 3 channels of asynchronous serial interfaces (UART).

This chapter describes asynchronous serial interface 0 (UART0). For UART1 and UART2, replace the SFR addresses and pin names as shown in Table 18-1 and Table 18-2.

Table 18-1 SFR Address Assignment

	UARTxCR1 (address)	UARTxCR2 (address)	UARTxDR (address)	UARTxSR (address)	RDxBUF (address)	TDxBUF (address)
UART0	UART0CR1 (0x0001A)	UART0CR2 (0x0001B)	UART0DR (0x0001C)	UART0SR (0x0001D)	RD0BUF (0x0001E)	TD0BUF (0x0001E)
UART1	UART1CR1 (0x00F54)	UART1CR2 (0x00F55)	UART1DR (0x00F56)	UART1SR (0x00F57)	RD1BUF (0x00F58)	TD1BUF (0x00F58)
UART2	UART2CR1 (0x00F5A)	UART2CR2 (0x00F5B)	UART2DR (0x00F5C)	UART2SR (0x00F5D)	RD2BUF (0x00F5E)	TD2BUF (0x00F5E)

Table 18-2 Pin Names

	Serial data input pin	Serial data output pin
UART0	RXD0 pin	TXD0 pin
UART1	RXD1 pin	TXD1 pin
UART2	RXD2 pin	TXD2 pin

Downloaded from Arrow.com.

18.1 Configuration

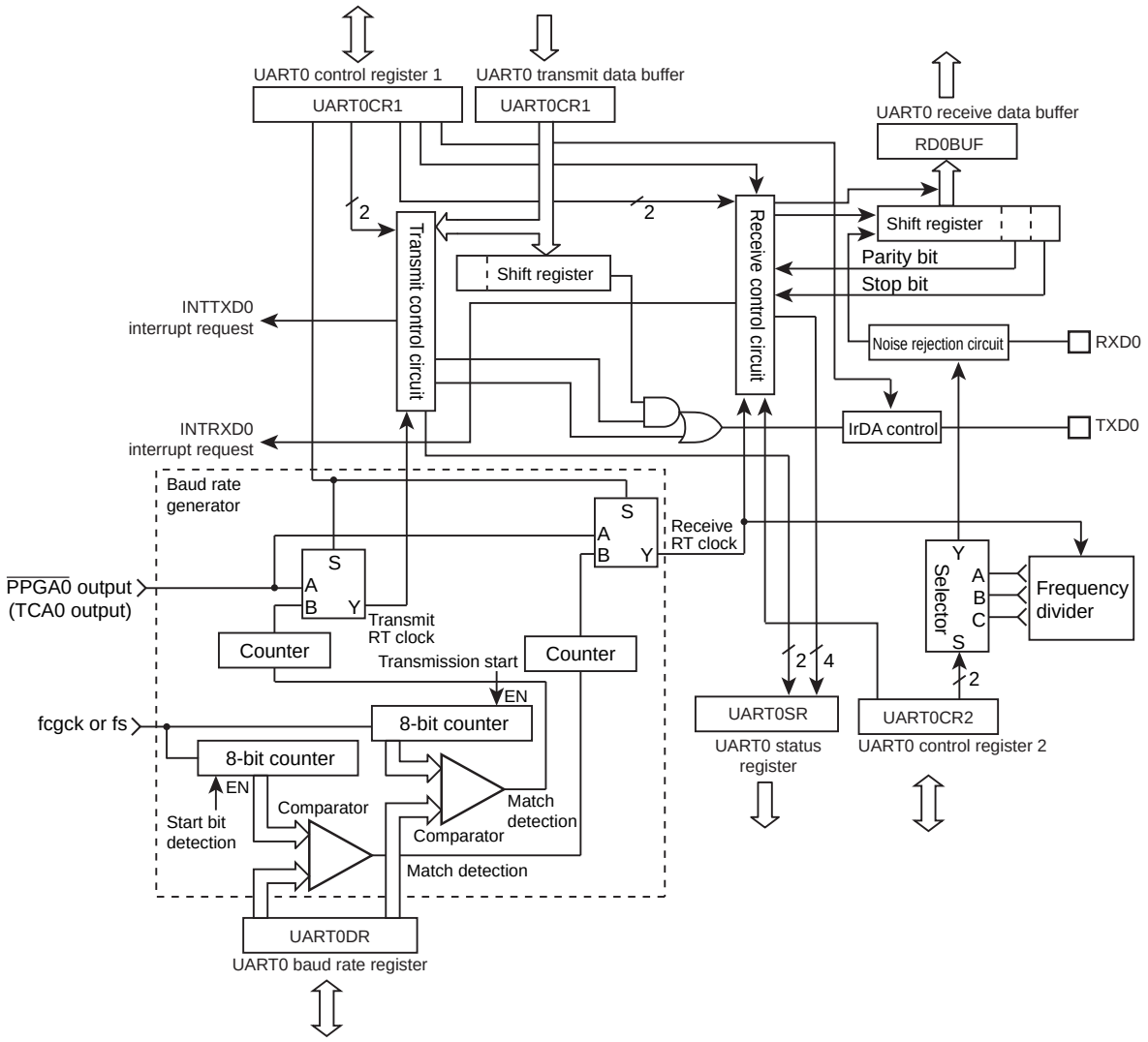


Figure 18-1 Asynchronous Serial Interface (UART)

Downloaded from Arrow.com.

18.2 Control

UART0 is controlled by the low power consumption registers (POFFCR1), UART0 control registers 1 and 2 (UART0CR1 and UART0CR2) and the UART0 baud rate register (UART0DR). The operating status can be monitored using the UART status register (UART0SR).

Low power consumption register 1

POFFCR1		7	6	5	4	3	2	1	0
(0x00F75)	Bit Symbol	-	-	-	SBI0EN	-	UART2EN	UART1EN	UART0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 Control	0 1	Disable Enable
UART2EN	UART2 Control	0 1	Disable Enable
UART1EN	UART1 Control	0 1	Disable Enable
UART0EN	UART0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

UART0 control register 1

UART0CR1		7	6	5	4	3	2	1	0
(0x0001A)	Bit Symbol	TXE	RXE	STOPBT	EVEN	PE	IRDASEL	BRG	-
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
	After reset	0	0	0	0	0	0	0	0

TXE	Transmit operation	0: 1:	Disable Enable	
RXE	Receive operation	0: 1:	Disable Enable	
STOPBT	Transmit stop bit length	0: 1:	1 bit 2 bits	
EVEN	Parity selection	0: 1:	Odd-numbered parity Even-numbered parity	
PE	Parity addition	0: 1:	No parity Parity added	
IRDASEL	TXD pin output selection	0: 1:	UART output IrDA output	
BRG	Transfer base clock selection		When SYSCR2<SYSCK> is "0"	When SYSCR2<SYSCK> is "1"
		0:	fcgck	fs
		1:	TCA0 output	

Note 1: fcgck, Gear clock; fs, Low-frequency clock

Note 2: If the TXE or RXE bit is set to "0" during the transmission or receiving of data, the operation is not disabled until the data transfer is completed. At this time, the data stored in the transmit data buffer is discarded.

Note 3: EVEN, PE and BRG settings are common to transmission and receiving.

Note 4: Set RXE and TXE to "0" before changing BRG.

Note 5: When BRG is set to the TCA0 output, the RT clock becomes asynchronous and the start bit of the transmitted/received data may get shorter by a maximum of (UART0DR+1)/(Transfer base clock frequency)[s].

If the pin is not used for the TCA0 output, control the TCA0 output by using the port function control register.

Note 6: To prevent STOPBT, EVEN, PE, IRDASEL and BRG from being changed accidentally during the UART communication, the register cannot be rewritten during the UART operation. For details, refer to "18.4 Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed ".

Note 7: When the STOP, IDLE0 or SLEEP0 mode is activated, TXE and RXE are cleared to "0" and the UART stops. Other bits keep their values.

Downloaded from Arrow.com.

UART0 control register 2

UART0CR2 (0x0001B)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	RTSEL			RXDNC		STOPBR
	Read/Write	R	R	R/W			R/W		R/W
	After reset	0	0	0	0	0	0	0	0

			Odd-numbered bits of transfer frame	Even-numbered bits of transfer frame
RTSEL	Selects the number of RT clocks	000:	16 clocks	16 clocks
		001:	16 clocks	17 clocks
		010:	15 clocks	15 clocks
		011:	15 clocks	16 clocks
		100:	17 clocks	17 clocks
		101:	Reserved	
		11*:	Reserved	
RXDNC	Selects the RXD input noise rejection time (Time of pulses to be removed as noise)	00:	No noise rejection	
		01:	1 x (UART0DR+1)/(Transfer base clock frequency) [s]	
		10:	2 x (UART0DR+1)/(Transfer base clock frequency) [s]	
		11:	4 x (UART0DR+1)/(Transfer base clock frequency) [s]	
STOPBR	Receive stop bit length	0:	1 bit	
		1:	2 bits	

Note 1: When a read instruction is executed on UART0CR2, bits 7 and 6 are read as "0".

Note 2: RTSEL can be set to two kinds of RT clocks for the even- and odd-numbered bits of the transfer frame. For details, refer to "18.8.1 Transfer baud rate calculation method".

Note 3: For details of the RXDNC noise rejection time, refer to "18.10 Received Data Noise Rejection".

Note 4: When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART0CR2 remains unchanged.

Note 5: When STOPBR is set to 2 bits, the first bit of the stop bits (during data receiving) is not checked for a framing error.

Note 6: To prevent RTSEL, RXDNC and STOPBR from being changed accidentally during the UART communication, the register cannot be rewritten during the UART operation. For details, refer to "18.4 Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed ".

UART0 baud rate register

UART0DR (0x0001C)		7	6	5	4	3	2	1	0
	Bit Symbol	UART0DR7	UART0DR6	UART0DR5	UART0DR4	UART0DR3	UART0DR2	UART0DR1	UART0DR0
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

Note 1: Set UART0CR1<RXE> and UART0CR1<TXE> to "0" before changing UART0DR. For the set values, refer to "18.8 Transfer Baud Rate".

Note 2: When UART0CR1<BRG> is set to the TCA0 output, the value set to UART0DR has no meaning.

Note 3: When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically but each bit value of UART0DR remains unchanged.

Downloaded from Arrow.com.

UART0 status register

UART0SR (0x0001D)	7	6	5	4	3	2	1	0
Bit Symbol	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

PERR	Parity error flag	0: 1:	No parity error Parity error
FERR	Framing error flag	0: 1:	No framing error Framing error
OERR	Overrun error flag	0: 1:	No overrrun error Overrun error
RBSY	Receive busy flag	0: 1:	Before receiving or end of receiving On receiving
RBFL	Receive buffer full flag	0: 1:	Receive buffer empty Receive buffer full
TBSY	Transmit busy flag	0: 1:	Before transmission or end of transmission On transmitting
TBFL	Transmit buffer full flag	0: 1:	Transmit buffer empty Transmit buffer full (Transmit data writing is completed)

Note 1: TBFL is cleared to "0" automatically after an INTTXD0 interrupt request is generated, and is set to "1" when data is set to TD0BUF.

Note 2: When a read instruction is executed on UART0SR, bit 4 is read as "0".

Note 3: When the STOP, IDLE0 or SLEEP0 mode is activated, each bit of UART0SR is cleared to "0" and the UART stops.

UART0 receive data buffer

RD0BUF (0x0001E)	7	6	5	4	3	2	1	0
Bit Symbol	RD0DR7	RD0DR6	RD0DR5	RD0DR4	RD0DR3	RD0DR2	RD0DR1	RD0DR0
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	0	0	0	0	0

Note 1: When the STOP, IDLE0 or SLEEP0 mode is activated, the RD0BUF values become undefined. If received data is required, read it before activating the mode.

UART0 transmit data buffer

TD0BUF (0x0001E)	7	6	5	4	3	2	1	0
Bit Symbol	TD0DR7	TD0DR6	TD0DR5	TD0DR4	TD0DR3	TD0DR2	TD0DR1	TD0DR0
Read/Write	W	W	W	W	W	W	W	W
After reset	0	0	0	0	0	0	0	0

Note 1: When the STOP, IDLE0 or SLEEP0 mode is activated, the TD0BUF values become undefined.

Downloaded from Arrow.com.

18.3 Low Power Consumption Function

UART0 has a low power consumption register (POFFCR1) that saves power consumption when the UART function is not used.

Setting POFFCR1<UART0EN> to "0" disables the basic clock supply to UART0 to save power. Note that this renders the UART unusable. Setting POFFCR1<UART0EN> to "1" enables the basic clock supply to UART0 and renders the UART usable.

After reset, POFFCR1<UART0EN> is initialized to "0", and this renders the UART unusable. When using the UART for the first time, be sure to set POFFCR1<UART0EN> to "1" in the initial setting of the program (before the UART control register is operated).

Do not change POFFCR1<UART0EN> to "0" during the UART operation, otherwise UART0 may operate unexpectedly.

Downloaded from Arrow.com.

18.4 Protection to Prevent UART0CR1 and UART0CR2 Registers from Being Changed

The TMP89FW24A has a function that protects the registers from being changed so that the UART communication settings (for example, stop bit and parity) are not changed accidentally during the UART operation.

Specific bits of UART0CR1 and UART0CR2 can be changed only under the conditions shown in Table 18-3. If a write instruction is executed on the register when it is protected from being changed, the bits remain unchanged and keep their previous values.

Table 18-3 Changing of UART0CR1 and UART0CR2

Bit to be changed	Function	Conditions that allow the bit to be changed			
		UART0CR1 <TXE>	UART0SR <TBSY>	UART0CR1 <RXE>	UART0SR <RBSY>
UART0CR1<STOPBT>	Transmit stop bit length	Both of these bits are "0"		-	-
UART0CR1<EVEN>	Parity selection	All of these bits are "0"			
UART0CR1<PE>	Parity addition				
UART0CR1<IRDASEL>	TXD pin output selection	Both of these bits are "0"		-	-
UART0CR1<BRG>	Transfer base clock selection	All of these bits are "0"			
UART0CR2<RTSEL>	Selection of number of RT clocks				
UART0CR2<RXDNC>	Selection of RXD pin input noise rejection time	-	-	Both of these bits are "0"	
UART0CR2<STOPBR>	Receive stop bit length				

2026-04-21
Downloaded from Arrow.com.

18.5 Activation of STOP, IDLE0 or SLEEP0 Mode

18.5.1 Transition of register status

When the STOP, IDLE0 or SLEEP0 mode is activated, the UART stops automatically and each register becomes the status as shown in Table 18-4. For the registers that do not hold their values, make settings again as needed after the operation mode is recovered.

Table 18-4 Transition of Register Status

	7	6	5	4	3	2	1	0
UART0CR1	TXE	RXE	STOPBT	EVEN	PE	IRDASEL	BRG	-
	Cleared to 0	Cleared to 0	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	-
UART0CR2	-	-	RTSEL			RXDNC		STOPBR
	-	-	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
UART0SR	PERR	FERR	OERR	-	RBSY	RBFL	TBSY	TBFL
	Cleared to 0	Cleared to 0	Cleared to 0	-	Cleared to 0	Cleared to 0	Cleared to 0	Cleared to 0
UART0DR	UART0DR7	UART0DR6	UART0DR5	UART0DR4	UART0DR3	UART0DR2	UART0DR1	UART0DR0
	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value	Hold the value
RD0BUF	RD0DR7	RD0DR6	RD0DR5	RD0DR4	RD0DR3	RD0DR2	RD0DR1	RD0DR0
	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate
TD0BUF	TD0DR7	TD0DR6	TD0DR5	TD0DR4	TD0DR3	TD0DR2	TD0DR1	TD0DR0
	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Indeterminate

18.5.2 Transition of TXD pin status

When the IDLE0, SLEEP0 or STOP mode is activated, the TXD pin reverts to the status shown in Table 18-5, whether data is transmitted/received or the operation is stopped.

Table 18-5 TXD Pin Status When the STOP, IDLE0 or SLEEP0 Mode Is Activated

UART0CR1 <IRDASEL>	IDLE0 or SLEEP0 mode	STOP mode	
		SYSCR1<OUTEN>="1"	SYSCR1<OUTEN>="0"
"0"	H level	H level	Hi-Z
"1"	L level	L level	

Downloaded from Arrow.com.

18.6 Transfer Data Format

The UART transfers data composed of the following four elements. The data from the start bit to the stop bit is collectively defined as a "transfer frame". The start bit consists of 1 bit (L level) and the data consists of 8 bits. Parity bits are determined by UART0CR1<PE> that selects the presence or absence of parity and UART0CR1<EVEN> that selects even- or odd-numbered parity. The bit length of the stop bit can be selected at UART0CR1<STBT>.

Figure 18-2 shows the transfer data format.

- Start bit (1 bit)
- Data (8 bits)
- Parity bit (selectable from even-numbered, odd-numbered or no parity)
- Stop bit (selectable from 1 bit or 2 bits)

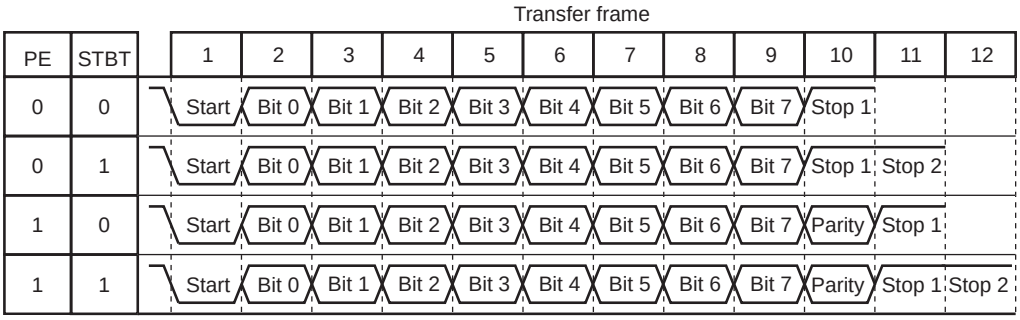


Figure 18-2 Transfer Data Format

18.7 Infrared Data Format Transfer Mode

The TXD0 pin can output data in the infrared data format (IrDA) by the setting of the IrDA output control register. Setting UART0CR1<IRDASEL> to "1" allows the TXD0 pin to output data in the infrared data format.

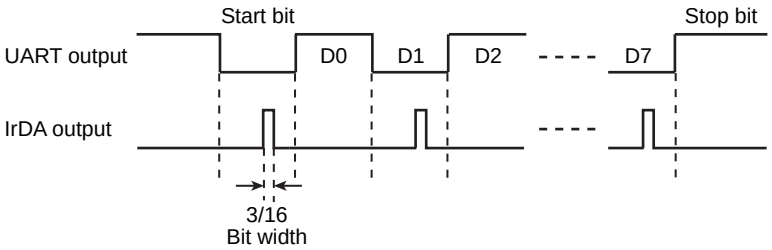


Figure 18-3 Example of Infrared Data Format (Comparison between Normal Output and IrDA Output)

Downloaded from Arrow.com.

18.8 Transfer Baud Rate

The transfer baud rate of UART is set by UART0CR1<BRG>, UART0DR and UART0CR2<RTSEL>. The settings of UART0DR and UART0CR2<RTSEL> for general baud rates and operating frequencies are shown below.

For independent calculation of transfer baud rates, refer to "18.8.1 Transfer baud rate calculation method".

Table 18-6 Set Values of UART0DR and UART0CR2<RTSEL> for Transfer Baud Rates (fcgck=10 to 1 MHz, UART0CR2<RXDNC>=0y00)

Basic baud rate [baud]	Register	Operating frequency										
		10MHz	8MHz	7.3728 MHz	6.144 MHz	6MHz	5MHz	4.9152 MHz	4.19MHz	4MHz	2MHz	1MHz
128000	UART0DR	0x04	0x03	-	0x02	0x02	-	-	0x01	0x01	0x00	-
	RTSEL	0y011	0y011	-	0y000	0y011	-	-	0y001	0y011	0y011	-
	Error	(+0.81%)	(+0.81%)	-	(0%)	(+0.81%)	-	-	(-0.80%)	(+0.81%)	(+0.81%)	-
115200	UART0DR	0x04	0x03	0x03	-	0x02	-	-	-	0x01	0x00	-
	RTSEL	0y100	0y100	0y000	-	0y100	-	-	-	0y100	0y100	-
	Error	(+2.12%)	(+2.12%)	(0%)	-	(+2.12%)	-	-	-	(+2.12%)	(+2.12%)	-
76800	UART0DR	0x07	0x06	0x05	0x04	0x04	0x03	0x03	-	0x02	-	-
	RTSEL	0y001	0y010	0y000	0y000	0y011	0y001	0y000	-	0y100	-	-
	Error	(-1.36%)	(-0.79%)	(0%)	(0%)	(+0.81%)	(-1.36%)	(0%)	-	(+2.12%)	-	-
62500	UART0DR	0x09	0x07	0x06	0x05	0x05	0x04	0x04	0x03	0x03	0x01	0x00
	RTSEL	0y000	0y000	0y100	0y001	0y000	0y000	0y011	0y100	0y000	0y000	0y000
	Error	(0%)	(0%)	(-0.87%)	(-0.70%)	(0%)	(0%)	(+1.48%)	(-1.41%)	(0%)	(0%)	(0%)
57600	UART0DR	0x0A	0x08	0x07	0x06	0x06	0x04	0x04	-	0x03	0x01	0x00
	RTSEL	0y000	0y011	0y000	0y010	0y010	0y100	0y100	-	0y100	0y100	0y100
	Error	(-1.36%)	(-0.44%)	(0%)	(+1.59%)	(-0.79%)	(+2.12%)	(+0.39%)	-	(+2.12%)	(+2.12%)	(+2.12%)
38400	UART0DR	0x10	0x0C	0x0B	0x09	0x09	0x07	0x07	0x06	0x06	0x02	-
	RTSEL	0y011	0y000	0y000	0y000	0y011	0y001	0y000	0y011	0y010	0y100	-
	Error	(-1.17%)	(+0.16%)	(0%)	(0%)	(+0.81%)	(-1.36%)	(0%)	(+0.57%)	(-0.79%)	(+2.12%)	-
19200	UART0DR	0x22	0x19	0x17	0x13	0x12	0x10	0x0F	0x0D	0x0C	0x06	0x02
	RTSEL	0y010	0y000	0y000	0y000	0y001	0y011	0y000	0y011	0y000	0y010	0y100
	Error	(-0.79%)	(+0.16%)	(0%)	(0%)	(-0.32%)	(-1.17%)	(0%)	(+0.57%)	(+0.16%)	(-0.79%)	(+2.12%)
9600	UART0DR	0x40	0x30	0x2F	0x27	0x26	0x22	0x1F	0x1C	0x19	0x0C	0x06
	RTSEL	0y000	0y100	0y000	0y000	0y000	0y010	0y000	0y010	0y000	0y000	0y010
	Error	(+0.16%)	(+0.04%)	(0%)	(0%)	(+0.16%)	(-0.79%)	(0%)	(+0.34%)	(+0.16%)	(+0.16%)	(-0.79%)
4800	UART0DR	0x8A	0x64	0x5F	0x4F	0x4D	0x40	0x3F	0x34	0x30	0x19	0x0C
	RTSEL	0y010	0y001	0y000	0y000	0y000	0y000	0y000	0y001	0y100	0y000	0y000
	Error	(-0.08%)	(+0.01%)	(0%)	(0%)	(+0.16%)	(+0.16%)	(0%)	(-0.18%)	(+0.04%)	(+0.16%)	(+0.16%)
2400	UART0DR	0xF4	0xC9	0xBF	0x9F	0x92	0x8A	0x7F	0x6C	0x64	0x30	0x19
	RTSEL	0y100	0y001	0y000	0y000	0y100	0y010	0y000	0y000	0y001	0y100	0y000
	Error	(+0.04%)	(+0.01%)	(0%)	(0%)	(+0.04%)	(-0.08%)	(0%)	(+0.11%)	(+0.01%)	(+0.04%)	(+0.16%)
1200	UART0DR	-	-	-	-	-	0xF4	0xFF	0xE8	0xC9	0x64	0x30
	RTSEL	-	-	-	-	-	0y100	0y000	0y010	0y001	0y001	0y100
	Error	-	-	-	-	-	(+0.04%)	(+0%)	(-0.10%)	(+0.01%)	(+0.01%)	(+0.04%)

Downloaded from Arrow.com.

Table 18-7 Set Values of UART0DR and UART0CR2<RTSEL> for Transfer Baud Rates (fs=32.768 kHz, UART0CR2<RXDNC>=0y00)

Basic baud rate [baud]	Register	Operating frequency
		32.768 kHz
300	UART0DR	0x06
	RTSEL	0y011
	Error	(+0.67%)
150	UART0DR	0x0D
	RTSEL	0y011
	Error	(+0.67%)
134	UART0DR	0x0E
	RTSEL	0y001
	Error	(-1.20%)
110	UART0DR	0x11
	RTSEL	0y001
	Error	(+0.30%)
75	UART0DR	0x1C
	RTSEL	0y010
	Error	(+0.44%)

Note 1: The overall error from the basic baud rate must be within ±3%. Even if the overall error is within ±3%, the communication may fail due to factors such as frequency errors in external controllers (for example, a personal computer) and oscillators and the load capacity of the communication pin.

18.8.1 Transfer baud rate calculation method

18.8.1.1 Bit width adjustment using UART0CR2<RTSEL>

The bit width of transmitted/received data can be finely adjusted by changing UART0CR2<RTSEL>. The number of RT clocks per bit can be changed in a range of 15 to 17 clocks by changing UART0CR2<RTSEL>. The RT clock is the transfer base clock, which is the pulses obtained by counting the clock selected at UART0CR1<BRG> the number of times of (UART0DR set value) + 1. Especially, when UART0CR2<RTSEL> is set to "0y001" or "0y011", two types of RT clocks alternate at each bit, so that the pseudo baud rates of RT × 15.5 clocks and RT × 16.5 clocks can be generated. The number of RT clocks per bit of transfer frame is shown in Figure 18-4.

For example, when fcgck is 4 [MHz], UART0CR2<RTSEL> is set to "0y000" and UART0DR is set to "0x19", the baud rate calculated using the formula in Figure 18-4 is expressed as:

fcgck / (16 × (UART0DR + 1) = 9615 [baud]

These settings generate a baud rate close to 9600 [baud] (+0.16%).

Downloaded from Arrow.com.

Transfer frame

PE	STBT	1	2	3	4	5	6	7	8	9	10	11	12
0	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1		
0	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop 1	Stop 2	
1	0	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	
1	1	Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Parity	Stop 1	Stop 2

RTSEL	Number of RT clocks												Generated baud rate
000	16	16	16	16	16	16	16	16	16	16	16	16	$\frac{fcgck}{16 \times (UARTDR+1)}$ [baud]
001	16	17	16	17	16	17	16	17	16	17	16	17	$\frac{fcgck}{16.5 \times (UARTDR+1)}$ [baud]
010	15	15	15	15	15	15	15	15	15	15	15	15	$\frac{fcgck}{15 \times (UARTDR+1)}$ [baud]
011	15	16	15	16	15	16	15	16	15	16	15	16	$\frac{fcgck}{15.5 \times (UARTDR+1)}$ [baud]
100	17	17	17	17	17	17	17	17	17	17	17	17	$\frac{fcgck}{17 \times (UARTDR+1)}$ [baud]

*When BRG is set to fcgck

Figure 18-4 Fine Adjustment of Baud Rate Clock Using UART0CR2<RTSEL>

18.8.1.2 Calculation of set values of UART0CR2<RTSEL> and UART0DR

The set value of UART0DR for an operating frequency and baud rate can be calculated using the calculation formula shown in Figure 18-5. For example, to generate a basic baud rate of 38400 [baud] with fcgck=4 [MHz], calculate the set value of UART0DR for each setting of UART0CR2<RTSEL> and compensate the calculated value to a positive number to obtain the generated baud rate as shown in Figure 18-6. Basically, select the set value of UART0CR2<RTSEL> that has the smallest baud rate error from among the generated baud rates. In Figure 18-6, the setting of UART0CR2<RTSEL>="0y010" has the smallest error among the calculated baud rates, and thus the generated baud rate is 38095 [baud] (−0.79%) against the basic baud rate of 38400 [baud].

Note: The error from the basic baud rate should be accurate to within ±3%. Even if the error is within ±3%, the communication may fail due to factors such as frequency errors of external controllers (for example, a personal computer) and oscillators and the load capacity of the communication pin.

RTSEL	UARTDR set value
000	$UARTDR = \frac{fcgck\ [Hz]}{16 \times A\ [baud]} - 1$
001	$UARTDR = \frac{fcgck\ [Hz]}{16.5 \times A\ [baud]} - 1$
010	$UARTDR = \frac{fcgck\ [Hz]}{15 \times A\ [baud]} - 1$
011	$UARTDR = \frac{fcgck\ [Hz]}{15.5 \times A\ [baud]} - 1$
100	$UARTDR = \frac{fcgck\ [Hz]}{17 \times A\ [baud]} - 1$

Figure 18-5 UART0DR Calculation Method (When BRG Is Set to fcgck)

Downloaded from Arrow.com.

RTSEL	UARTDR calculation	Generated baud rate
000	$\text{UARTDR} = \frac{4000000\ [\text{Hz}]}{16 \times 38400\ [\text{baud}]} - 1 \approx 6$	$\frac{4000000\ [\text{Hz}]}{16 \times (6 + 1)} = 35714\ [\text{baud}]\ (-6.99\%)$
001	$\text{UARTDR} = \frac{4000000\ [\text{Hz}]}{16.5 \times 38400\ [\text{baud}]} - 1 \approx 5$	$\frac{4000000\ [\text{Hz}]}{16.5 \times (5 + 1)} = 40404\ [\text{baud}]\ (+5.22\%)$
010	$\text{UARTDR} = \frac{4000000\ [\text{Hz}]}{15 \times 38400\ [\text{baud}]} - 1 \approx 6$	$\frac{4000000\ [\text{Hz}]}{15 \times (6 + 1)} = 38095\ [\text{baud}]\ (-0.79\%)$
011	$\text{UARTDR} = \frac{4000000\ [\text{Hz}]}{15.5 \times 38400\ [\text{baud}]} - 1 \approx 6$	$\frac{4000000\ [\text{Hz}]}{15.5 \times (6 + 1)} = 36866\ [\text{baud}]\ (-3.99\%)$
100	$\text{UARTDR} = \frac{4000000\ [\text{Hz}]}{17 \times 38400\ [\text{baud}]} - 1 \approx 5$	$\frac{4000000\ [\text{Hz}]}{17 \times (5 + 1)} = 39216\ [\text{baud}]\ (+2.12\%)$

Figure 18-6 Example of UART0DR Calculation

Downloaded from Arrow.com.

18.9 Data Sampling Method

The UART receive control circuit starts RT clock counting when it detects a falling edge of the input pulses to the RXD0 pin. 15 to 17 RT clocks are counted per bit and each clock is expressed as RTn (n=16 to 0). In a bit that has 17 RT clocks, RT16 to RT0 are counted. In a bit that has 16 RT clocks, RT15 to RT0 are counted. In a bit that has 15 RT clocks, RT14 to RT0 are counted (Decrement). During counting of RT8 to RT6, the UART receive control circuit samples the input pulses to the RXD0 pin to make a majority decision. The same level detected twice or more from among three samplings is processed as the data for the bit.

The number of RT clocks can be changed in a range of 15 to 17 by setting UART0CR2<RTSEL>. However, sampling is always executed in RT8 to RT6, even if the number of RT clocks is changed (Figure 18-7).

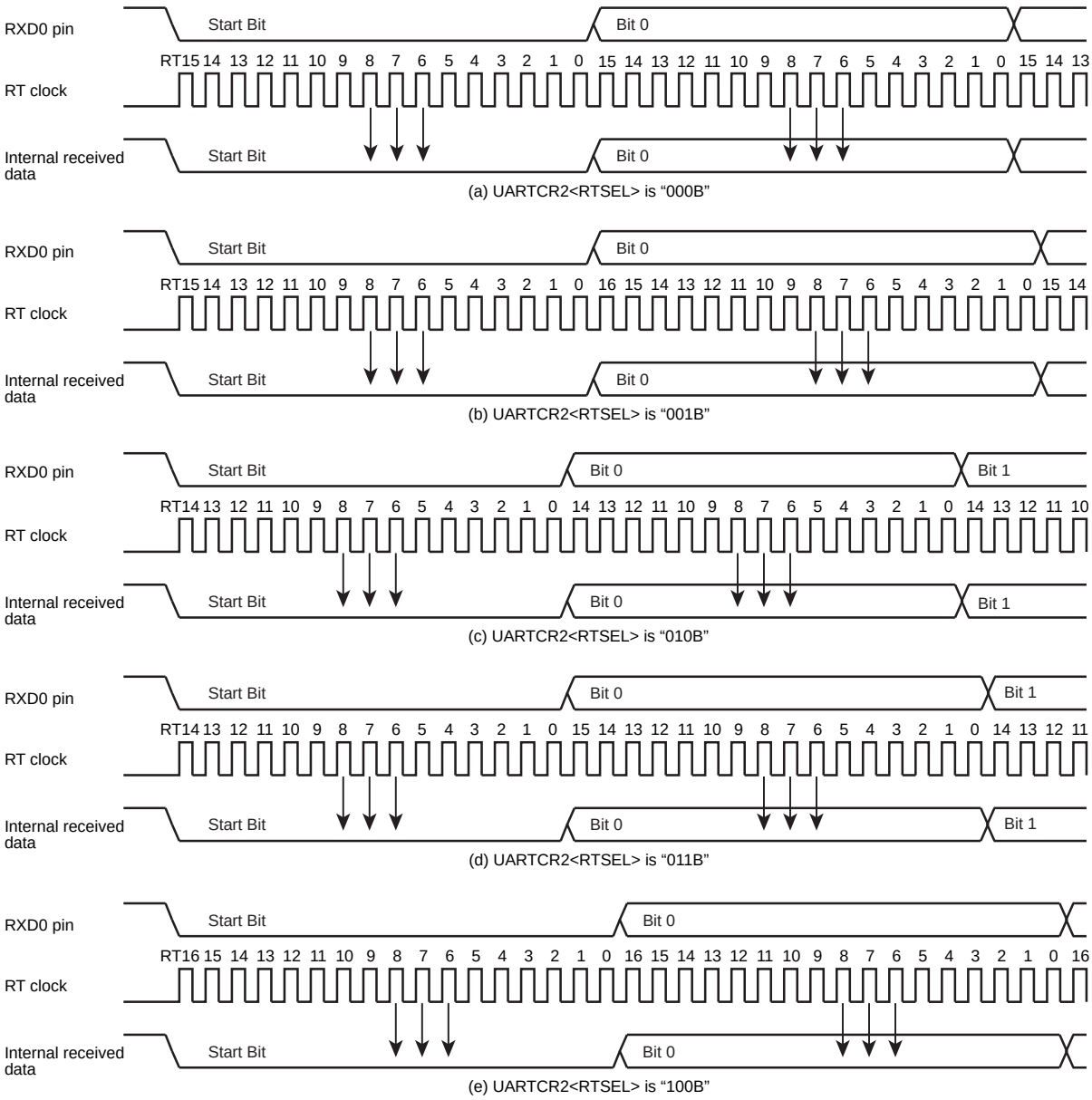


Figure 18-7 Data Sampling in Each Case of UARTCR2<RTSEL>

Downloaded from Arrow.com.

If "1" is detected in sampling of the start bit, for example, due to the influence of noise, RT clock counting stops and the data receiving is suspended. Subsequently, when a falling edge is detected in the input pulses to the RXD0 pin, RT clock counting restarts and the data receiving restarts with the start bit.

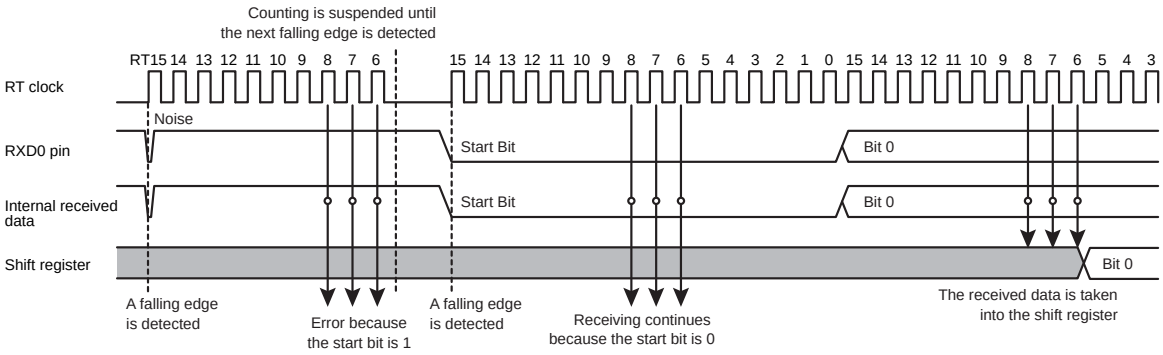


Figure 18-8 Start Bit Sampling

Downloaded from Arrow.com.

18.10 Received Data Noise Rejection

When noise rejection is enabled at UART0CR2<RXDNC>, the time of pulses to be regarded as signals is as shown in Table 18-8.

Table 18-8 Received Data Noise Rejection Time

RXDNC	Noise rejection time [s]	Time of pulses to be regarded as signals
00	No noise rejection	-
01	(UART0DR+1)/(Transfer base clock frequency)	2 × (UART0DR+1)/(Transfer base clock frequency)
10	2 × (UART0DR+1)/(Transfer base clock frequency)	4 × (UART0DR+1)/(Transfer base clock frequency)
11	4 × (UART0DR+1)/(Transfer base clock frequency)	8 × (UART0DR+1)/(Transfer base clock frequency)

Note 1: The transfer base clock frequency is the clock frequency selected at UARTCR1<BRG>.

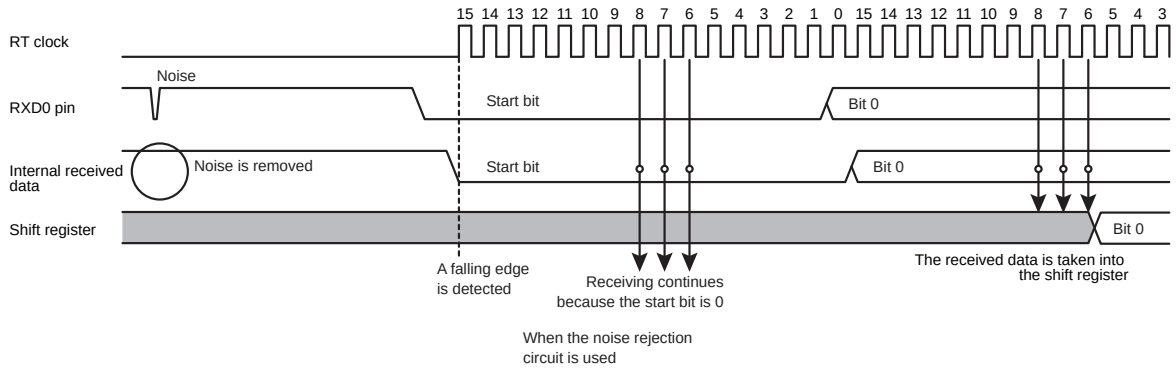


Figure 18-9 Received Data Noise Rejection

Downloaded from Arrow.com.

18.11 Transmit/Receive Operation

18.11.1 Data transmit operation

Set UART0CR1<TXE> to "1". Check UART0SR<TBFL> = "0", and then write data into TD0BUF (transmit data buffer). Writing data into TD0BUF sets UART0SR<TBFL> to "1", transfers the data to the transmit shift register, and outputs the data sequentially from the TXD0 pin. The data output includes a start bit, stop bits whose number is specified in UART0CR1<STBT> and a parity bit if parity addition is specified. Select the data transfer baud rate using UART0CR1<BRG>, UART0CR2<RTSEL> and UART0DR. When data transmission starts, the transmit buffer full flag UART0SR<TBFL> is cleared to "0" and an INTTXD0 interrupt request is generated.

Note 1: After data is written into TD0BUF, if new data is written into TD0BUF before the previous data is transferred to the shift register, the new data is written over the previous data and is transferred to the shift register.

Note 2: Under the conditions shown in Table 18-9, the TXD0 pin output is fixed at the L or H level according to the setting of UART0CR1<IRDASEL>.

Table 18-9 TXD0 Pin Output

Condition	TXD0 pin output	
	IRDASEL="0"	IRDASEL="1"
When UART0CR1<TXE> is "0"	H level	L level
From when "1" is written to UART0CR1<TXE> to when the transmitted data is written to TD0BUF		
When the STOP, IDLE0 or SLEEP0 mode is active		

18.11.2 Data receive operation

Set UART0CR1<RXE> to "1". When data is received via the RXD0 pin, the received data is transferred to RD0BUF (receive data buffer). At this time, the transmitted data includes a start bit, stop bit(s) and a parity bit if parity addition is specified. When the stop bit(s) are received, data only is extracted and transferred to RD0BUF (receive data buffer). Then the receive buffer full flag UART0SR<RBFL> is set and an INTRXD0 interrupt request is generated. Set the data transfer baud rate using UART0CR1<BRG>, UART0CR2<RTSEL> and UART0DR.

If an overrun error occurs when data is received, the data is not transferred to RD0BUF (receive data buffer) but discarded; data in the RD0BUF is not affected.

Downloaded from Arrow.com.

18.12 Status Flag

18.12.1 Parity error

When the parity determined using the receive data bits differs from the received parity bit, the parity error flag UART0SR<PERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

If UART0SR<PERR> is "1" when UART0SR is read, UART0SR<PERR> will be cleared to "0" when RD0BUF is read subsequently. (The RD0BUF read value becomes undefined.)

If UART0SR<PERR> is set to "1" after UART0SR is read, UART0SR<PERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<PERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

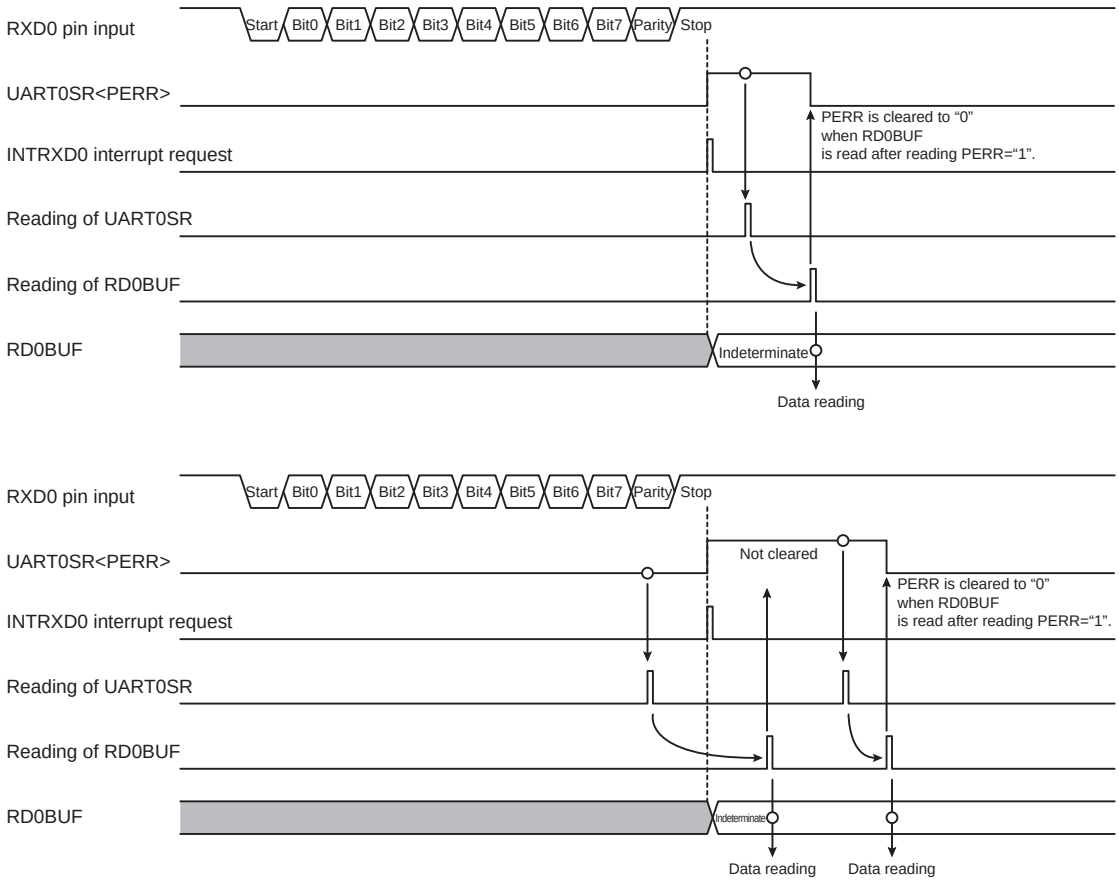


Figure 18-10 Occurrence of Parity Error

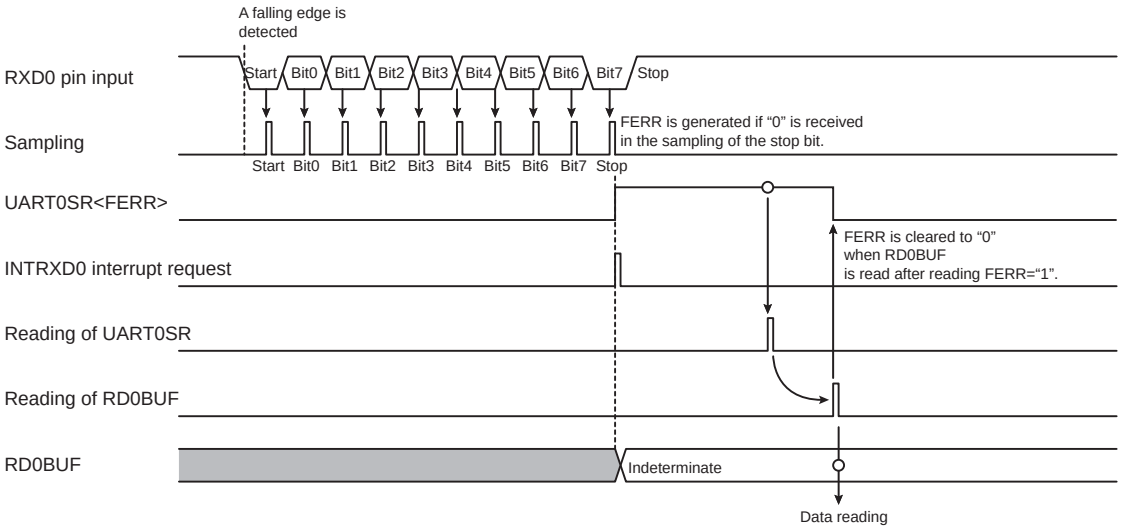
Downloaded from Arrow.com.

18.12.2 Framing Error

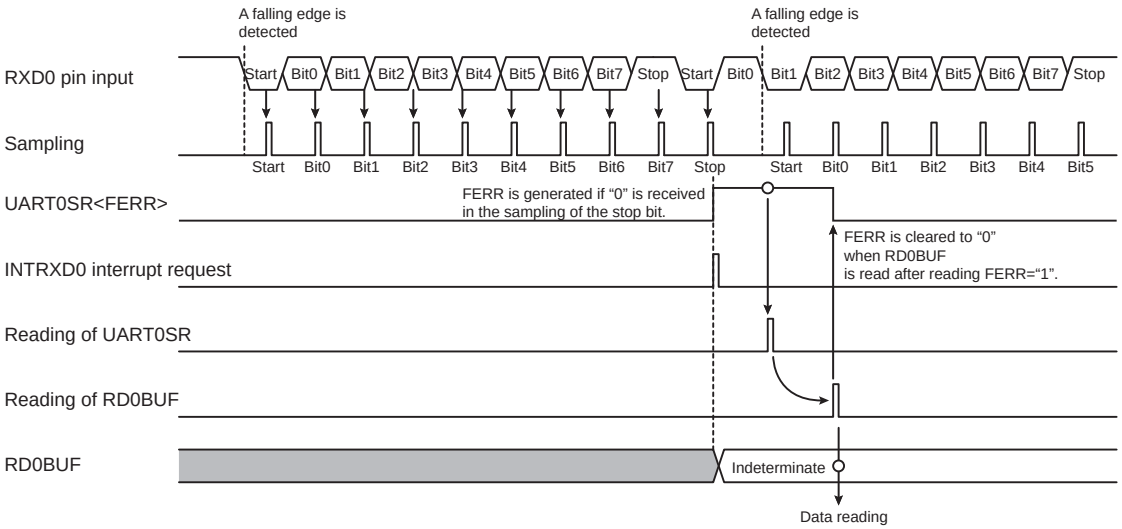
If the internal and external baud rates differ or "0" is sampled as the stop bit of received data due to the influence of noise on the RXD0 pin, the framing error flag UART0SR<FERR> is set to "1". At this time, an INTRXD0 interrupt request is generated.

If UART0SR<FERR> is "1" when UART0SR is read, UART0SR<FERR> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR<FERR> is set to "1" after UART0SR is read, UART0SR<FERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<FERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read.



When the external baud rate is slower than the internally set baud rate



When the external baud rate is faster than the internally set baud rate

Figure 18-11 Occurrence of Framing Error

Downloaded from Arrow.com.

18.12.3 Overrun error

If receiving of all data bits is completed before the previous received data is read from RD0BUF, the overrun error flag UART0SR<OERR> is set to "1" and an INTRXD0 interrupt request is generated. The data received at the occurrence of the overrun error is discarded and the previous received data is maintained. Subsequently, if data is received while UART0SR<OERR> is still "1", no INTRXD0 interrupt request is generated, and the received data is discarded. (Figure 18-12)

Note that parity or framing errors in the discarded received data cannot be detected. (These error flags are not set.) That is to say, if these errors are detected together with an overrun error during the reading of UART0SR, they have occurred in the previous received data (the data stored in RD0BUF). (Figure 18-13)

If UART0SR<OERR> is "1" when UART0SR is read, UART0SR<OERR> will be cleared to "0" when RD0BUF is read subsequently. (Figure 18-14)

If UART0SR<OERR> is set to "1" after UART0SR is read, UART0SR<OERR> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<OERR> will be cleared to "0" when UART0SR is read again and RD0BUF is read. (Figure 18-14)

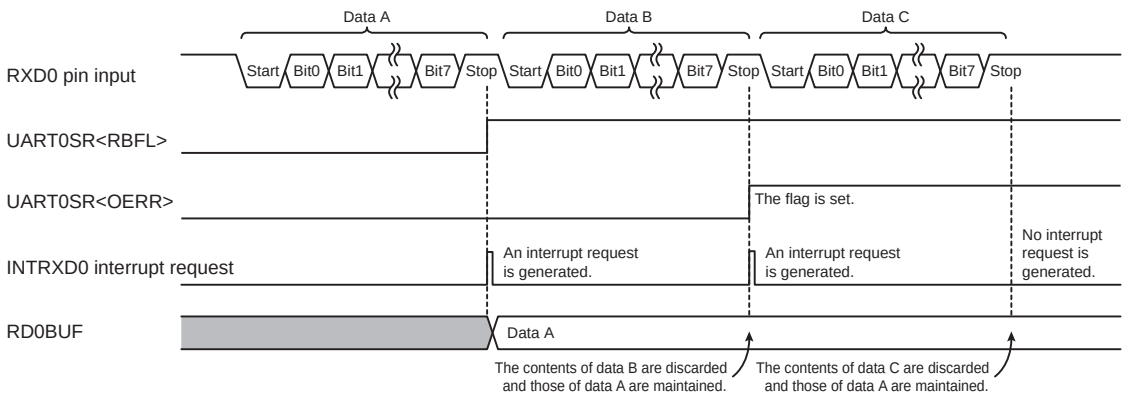
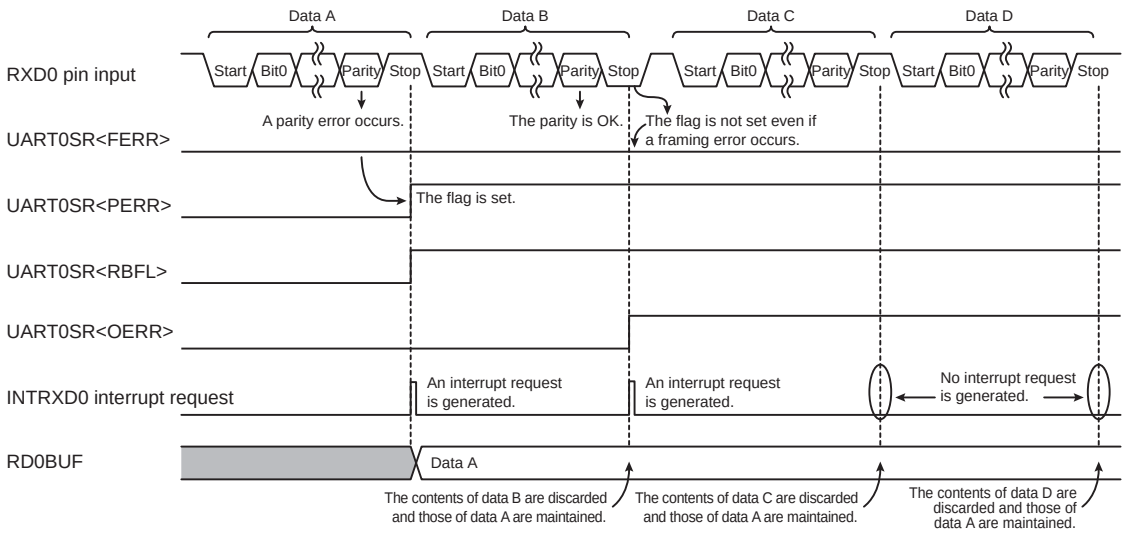
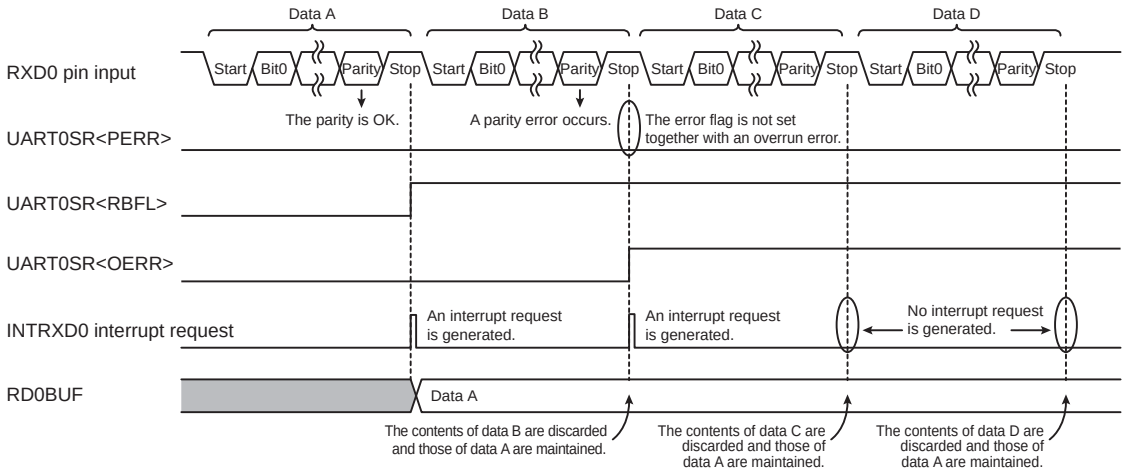


Figure 18-12 Generation of INTRXD0 Interrupt Request

Downloaded from Arrow.com.



When a parity error occurs in the first received data and a framing error occurs in the second data



When a parity error occurs in the second received data

Figure 18-13 Framing/Parity Error Flags When an Overrun Error Occurs

Downloaded from Arrow.com.

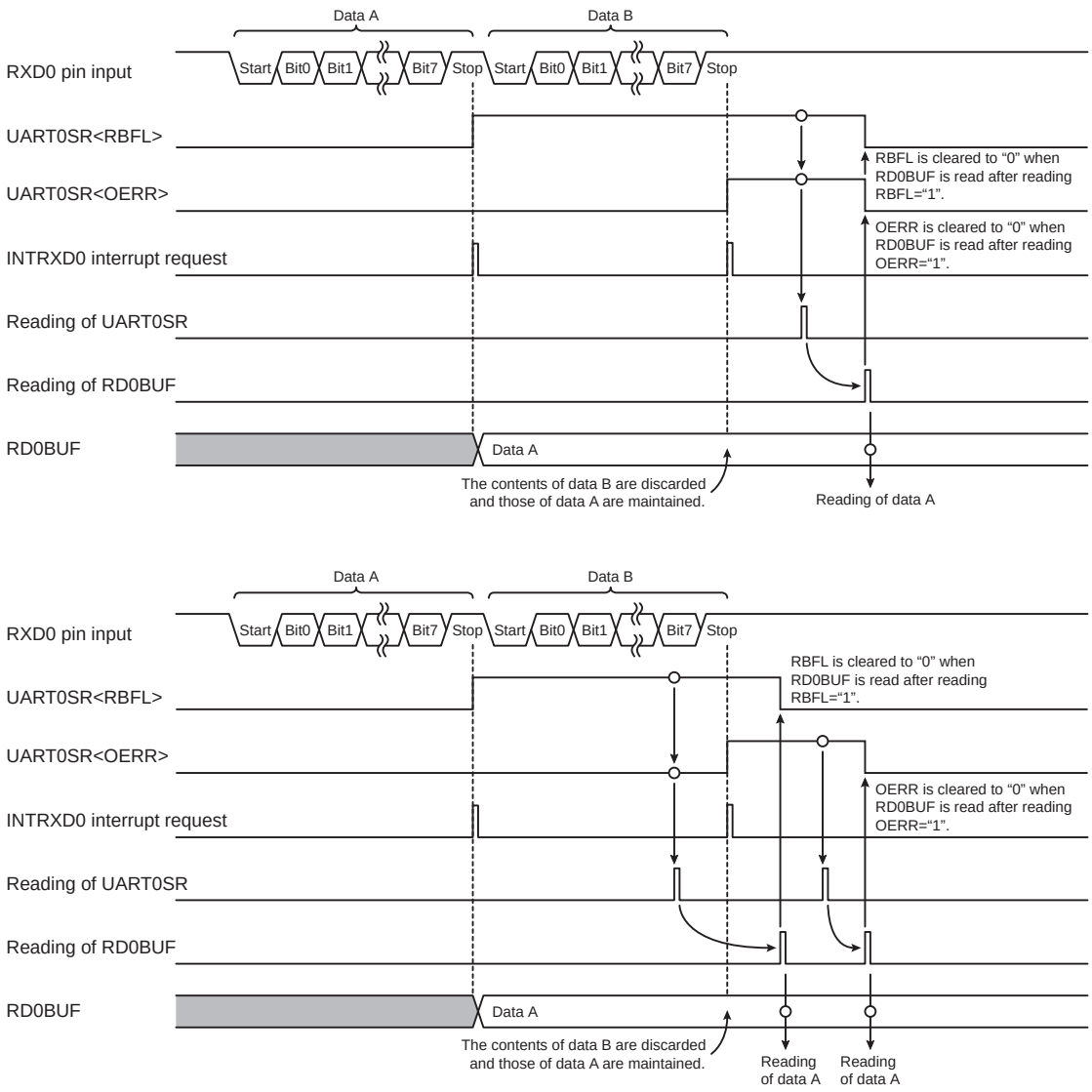


Figure 18-14 Clearance of Overrun Error Flag

Downloaded from Arrow.com.

18.12.4 Receive Data Buffer Full

Loading the received data in RD0BUF sets UART0SR<RBFL> to "1".

If UART0SR<RBFL> is "1" when UART0SR is read, UART0SR<RBFL> will be cleared to "0" when RD0BUF is read subsequently.

If UART0SR<RBFL> is set to "1" after UART0SR is read, UART0SR<RBFL> will not be cleared to "0" when RD0BUF is read subsequently. In this case, UART0SR<RBFL> will be cleared to "0" when UART0SR is read again and RD0BUF is read.

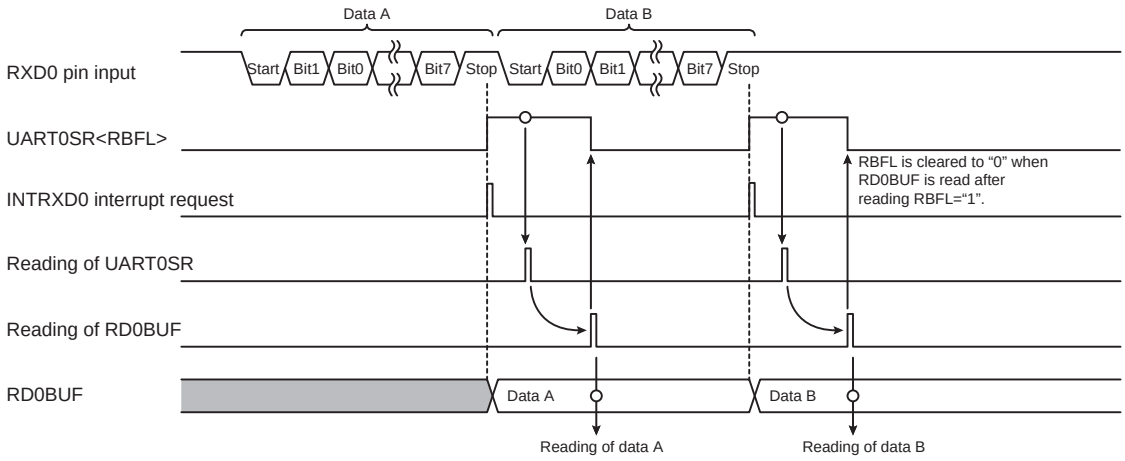


Figure 18-15 Occurrence of Receive Data Buffer Full

Downloaded from Arrow.com.

18.12.5 Transmit busy flag

If transmission is completed with no waiting data in TD0BUF (when UART0SR<TBFL>="0"), UART0SR<TBSY> is cleared to "0". When transmission is restarted after data is written into TD0BUF, UART0SR<TBSY> is set to "1". At this time, an INTTXD0 interrupt request is generated.

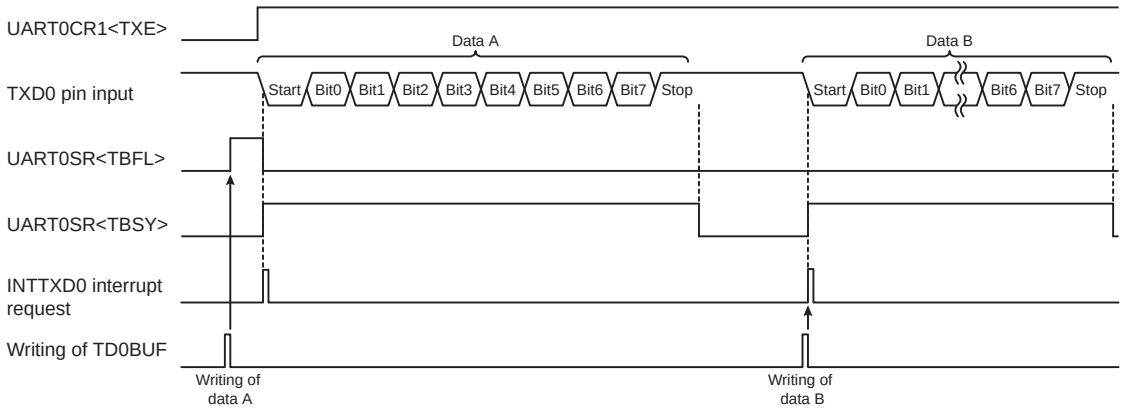


Figure 18-16 Transmit Busy Flag and Occurrence of Transmit Buffer Full

18.12.6 Transmit Buffer Full

When TD0BUF has no data, or when data in TD0BUF is transferred to the transmit shift register and transmission is started, UART0SR<TBFL> is cleared to "0". At this time, an INTTXD0 interrupt request is generated.

Writing data into TD0BUF sets UART0SR<TBFL> to "1".

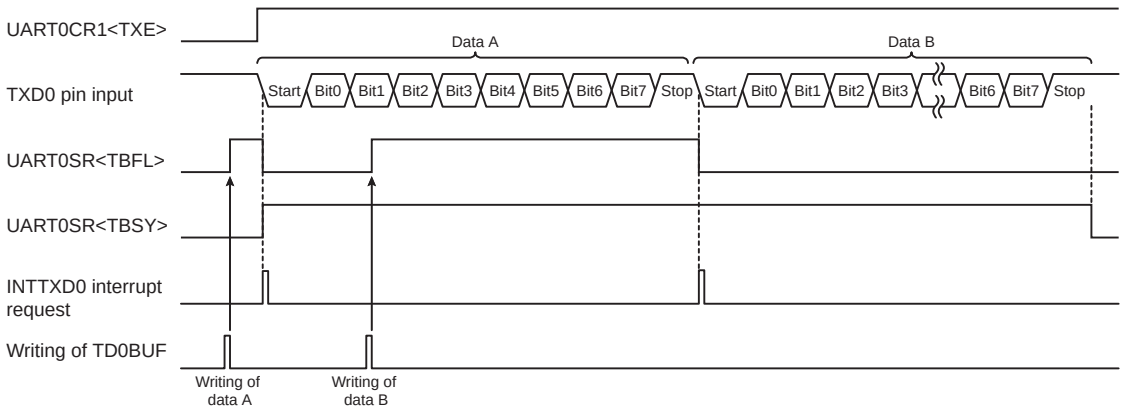


Figure 18-17 Occurrence of Transmit Buffer Full

Downloaded from Arrow.com.

18.13 Receiving Process

Figure 18-18 shows an example of the receiving process. Details of flag judgments in the processing are shown in Table 18-10 and Table 18-11.

If any framing error or parity error is detected, the received data has erroneous value(s). Execute the error handling, for example, by discarding the received data read from RD0BUF and receiving the data again.

If any overrun error is detected, the receiving of one or more pieces of data is unfinished. It is impossible to determine the number of pieces of data that could not be received. Execute the error handling, for example, by receiving data again from the beginning of the transfer. Basically, an overrun error occurs when the internal software processing cannot follow the data transfer speed. It is recommended to slow the transfer baud rate or modify the software to execute flow control.

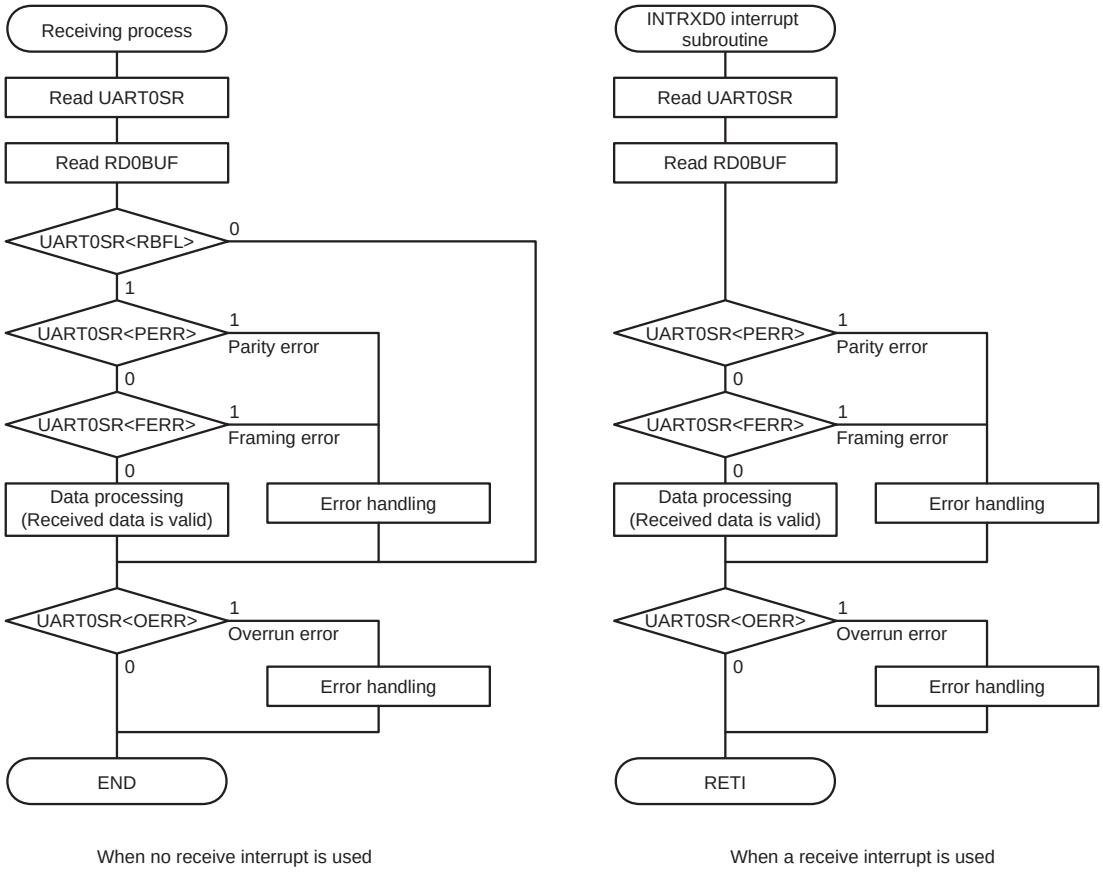


Figure 18-18 Example of Receiving Process

Note 1: If multiple interrupts are used in the INTRXD0 interrupt subroutine, the interrupt should be enabled after reading UART0SR and RD0BUF.

Downloaded from Arrow.com.

Table 18-10 Flag Judgments When No Receive Interrupt Is Used

RBFL	FERR/PERR	OERR	State
0	-	0	Data has not been received yet.
0	-	1	Some pieces of data could not be received during the previous data receiving process (Receiving of next data is completed in the period from when UART0SR is read to when RD0BUF is read in the previous data receiving process.)
1	0	0	Receiving has been completed properly.
1	0	1	Receiving has been completed properly, but some pieces of data could not be received.
1	1	0	Received data has erroneous value(s).
1	1	1	Received data has erroneous value(s) and some pieces of data could not be received.

Table 18-11 Flag Judgments When a Receive Interrupt Is Used

FERR/PERR	OERR	State
0	0	Receiving has been completed properly.
0	1	Receiving has been completed properly, but some pieces of data could not be received.
1	0	Received data has erroneous value(s).
1	1	Received data has erroneous value(s) and some pieces of data could not be received.

Downloaded from Arrow.com.

18.14 AC Properties

18.14.1 IrDA properties

(V_{SS} = 0 V, Topr = −40 to 85°C)

Item	Condition	Min	Typ.	Max	Unit
TXD output pulse time (RT clock × (3/16))	Transfer baud rate = 2400 bps	-	78.13	-	μs
	Transfer baud rate = 9600 bps	-	19.53	-	
	Transfer baud rate = 19200 bps	-	9.77	-	
	Transfer baud rate = 38400 bps	-	4.88	-	
	Transfer baud rate = 57600 bps	-	3.26	-	
	Transfer baud rate = 115200 bps	-	1.63	-	

Downloaded from Arrow.com.

19. Synchronous Serial Interface (SIO)

The TMP89FW24A contains 2 channels of high-speed 8-bit serial interfaces of the clock synchronization type.

This chapter describes serial interface 0. For serial interface 1, replace the SFR addresses and pin names as shown in Table 19-1 and Table 19-2.

Table 19-1 SFR Address Assignment

	SIOxCR (address)	SIOxSR (address)	SIOxBUF (address)
Serial interface 0	SIO0CR (0x0001F)	SIO0SR (0x00020)	SIO0BUF (0x00021)
Serial interface 1	SIO1CR (0x00F70)	SIO1SR (0x00F71)	SIO1BUF (0x00F72)

Table 19-2 Pin Names

	Serial clock input/output pin	Serial data input pin	Serial data output pin
Serial interface 0	SCLK0 pin	SI0 pin	SO0 pin
Serial interface 1	SCLK1 pin	SI1 pin	SO1 pin

Downloaded from Arrow.com.

19.1 Configuration

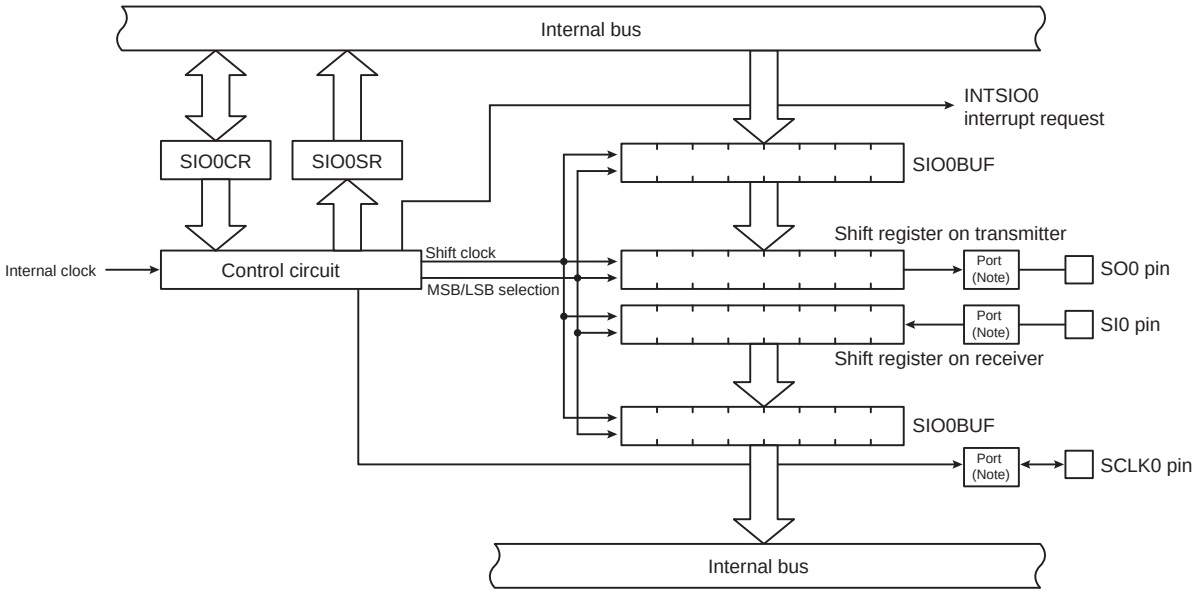


Figure 19-1 Serial Interface

Note: The serial interface input/output pins are also used as the I/O ports. The I/O port register settings are required to use these pins for a serial interface. For details, refer to the chapter of I/O ports.

Downloaded from Arrow.com.

19.2 Control

The synchronous serial interface SIO0 is controlled by the low power consumption registers (POFFCR2), the serial interface data buffer register (SIO0BUF), the serial interface control register (SIO0CR) and the serial interface status register (SIO0SR).

Low power consumption register 2

POFFCR2 (0x00F76)	7	6	5	4	3	2	1	0
Bit Symbol	LCDEN	-	RTCEN	-	-	-	SIO1EN	SIO0EN
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

LCDEN	LCD Control	0 1	Disable Enable
RTCEN	RTC Control	0 1	Disable Enable
SIO1EN	SIO1 Control	0 1	Disable Enable
SIO0EN	SIO0 Control	0 1	Disable Enable

Serial interface buffer register

SIO0BUF (0x00021)	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF							
Read/Write	R							
After reset	0	0	0	0	0	0	0	0

Serial interface buffer register

SIO0BUF (0x00021)	7	6	5	4	3	2	1	0
Bit Symbol	SIO0BUF							
Read/Write	W							
After reset	1	1	1	1	1	1	1	1

Note 1: SIO0BUF is the data buffer for both transmission and reception. The last received data is read each time SIO0BUF is read. If SIO0BUF has never received data, it is read as "0". When data is written into it, the data is treated as the transmit data.

Downloaded from Arrow.com.

Serial interface control register

SIO0CR (0x0001F)		7	6	5	4	3	2	1	0
	Bit Symbol	SIOEDG	SIOCKS			SIODIR	SIOS	SIOM	
	Read/Write	R/W	R/W			R/W	R/W	R/W	
	After reset	0	0	0	0	0	0	0	0

SIOEDG	Transfer edge selection	0	0: Receive data at a rising edge and transmit data at a falling edge	
		1	1: Transmit data at a rising edge and receive data at a falling edge	
SIOCKS	Serial clock selection [Hz]		NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode
		000	fcgck/2^9	-
		001	fcgck/2^6	-
		010	fcgck/2^5	-
		011	fcgck/2^4	-
		100	fcgck/2^3	-
		101	fcgck/2^2	-
		110	fcgck/2	fs/2^3
		111	External clock input	
SIODIR	Transfer format (MSB/LSB) selection	0	LSB first (transfer from bit 0)	
		1	MSB first (transfer from bit 7)	
SIOS	Transfer operation start/stop instruction	0	0: Operation stop (reserved stop)	
		1	1: Operation start	
SIOM	Transfer mode selection and operation	00	Operation stop (forced stop)	
		01	8-bit transmit mode	
		10	8-bit receive mode	
		11	8-bit transmit and receive mode	

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock [Hz]

Note 2: After the operation is started by writing "1" to SIOS, writing to SIOEDG, SIOCKS and SIODIR is invalid until SIO0SR<SIOF> becomes "0". (SIOEDG, SIOCKS and SIODIR can be changed at the same time as changing SIOS from "0" to "1".)

Note 3: After the operation is started by writing "1" to SIOS, no values other than"00" can be written to SIOM until SIOF becomes "0" (if a value from "01" to "11" is written to SIOM, it is ignored). The transfer mode cannot be changed during the operation.

Note 4: SIOS remains at "0", if "1" is written to SIOS when SIOM is "00" (operation stop).

Note 5: When SIO is used in SLOW1/2 or SLEEP1 mode, be sure to set SIOCKS to "110". If SIOCKS is set to any other value, SIO will not operate. When SIO is used in SLOW1/2 or SLEEP1 mode, execute communications with SIOCKS="110" in advance or change SIOCKS after SIO is stopped.

Note 6: When STOP, IDLE0 or SLEEP0 mode is activated, SIOM is automatically cleared to "00" and SIO stops the operation. At the same time, SIOS is cleared to "0". However, the values set for SIOEDG, SIOCKS and SIODIR are maintained.

Downloaded from Arrow.com.

Serial interface status register

SIO0SR		7	6	5	4	3	2	1	0
(0x00020)	Bit Symbol	SIOF	SEF	OERR	REND	UERR	TBFL	-	-
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

SIOF	Serial transfer operation status monitor	0 1	Transfer not in progress Transfer in progress
SEF	Shift operation status monitor	0 1	Shift operation not in progress Shift operation in progress
OERR	Receive overrun error flag	0 1	No overrun error has occurred At least one overrun error has occurred
REND	Receive completion flag	0 1	No data has been received since the last receive data was read out At least one data receive operation has been executed
UERR	Transmit underrun error flag	0 1	No transmit underrun error has occurred At least one transmit underrun error has occurred
TBFL	Transmit buffer full flag	0 1	The transmit buffer is empty The transmit buffer has the data that has not yet been transmitted

Note 1: The OERR and UERR flags are cleared by reading SIO0SR.

Note 2: The REND flag is cleared by reading SIO0BUF.

Note 3: Writing "00" to SIO0CR<SIOM> clears all the bits of SIO0SR to "0", whether the serial interface is operating or not. When STOP, IDLE0 or SLEEP0 mode is activated, SIOM is automatically cleared to "00" and all the bits of SIO0SR are cleared to "0".

Note 4: Bit 1 to 0 of SIO0SR are read "0".

Downloaded from Arrow.com.

19.3 Low Power Consumption Function

Serial interface 0 has the low power consumption registers (POFFCR2) that save power when the serial interface is not being used.

Setting POFFCR2<SIO0EN> to "0" disables the basic clock supply to serial interface 0 to save power. Note that this renders the serial interface unusable. Setting POFFCR2<SIO0EN> to "1" enables the basic clock supply to serial interface 0 and allows the serial interface to operate.

After reset, POFFCR2<SIO0EN> are initialized to "0", and this renders the serial interface unusable. When using the serial interface for the first time, be sure to set POFFCR2<SIO0EN> to "1" in the initial setting of the program (before the serial interface control registers are operated).

Do not change POFFCR2<SIO0EN> to "0" during the serial interface operation. Otherwise serial interface 0 may operate unexpectedly.

Downloaded from Arrow.com.

19.4 Functions

19.4.1 Transfer format

The transfer format can be set to either MSB or LSB first by using SIO0CR<SIODIR>. Setting SIO0CR<SIODIR> to "0" selects LSB first as the transfer format. In this case, the serial data is transferred in sequence from the least significant bit.

Setting SIO0CR<SIODIR> to "1" selects MSB first as the transfer format. In this case, the serial data is transferred in sequence from the most significant bit.

19.4.2 Serial clock

The serial clock can be selected by using SIO0CR<SIOCKS>.

Setting SIO0CR<SIOCKS> to "000" to "110" selects the internal clock as the serial clock. In this case, the serial clock is output from the SCLK0 pin. The serial data is transferred in synchronization with the edge of the SCLK0 pin output.

Setting SIO0CR<SIOCKS> to "111" selects an external clock as the serial clock. In this case, an external serial clock must be input to the SCLK0 pin. The serial data is transferred in synchronization with the edge of the external clock.

The serial data transfer edge can be selected for both the external and internal clocks. For details, refer to "19.4.3 Transfer edge selection".

Table 19-3 Transfer Baud Rate

SIO0CR <SIOCKS>	Serial clock [Hz]		fcgck=4MHz		fcgck=8MHz		fcgck=10MHz		fs=32.768kHz	
	NORMAL1/2 or IDLE1/2 mode	SLOW1/2 or SLEEP1 mode	1-bit time (μs)	Baud rate (bps)	1-bit time (μs)	Baud rate (bps)	1-bit time (μs)	Baud rate (bps)	1-bit time (μs)	Baud rate (bps)
000	$fcgck/2^9$	-	128	7.813k	64	15.625k	51.2	19.531k	-	-
001	$fcgck/2^6$	-	16	62.5k	8	125k	6.4	156.25k	-	-
010	$fcgck/2^5$	-	8	125k	4	250k	3.2	312.5k	-	-
011	$fcgck/2^4$	-	4	250k	2	500k	1.6	625k	-	-
100	$fcgck/2^3$	-	2	500k	1	1M	0.8	1.25M	-	-
101	$fcgck/2^2$	-	1	1M	0.5	2M	0.4	2.5M	-	-
110	fcgck/2	$fs/2^3$	0.5	2M	0.25	4M	0.2	5M	244	4k

19.4.3 Transfer edge selection

The serial data transfer edge can be selected by using SIOCR<SIOEDG>.

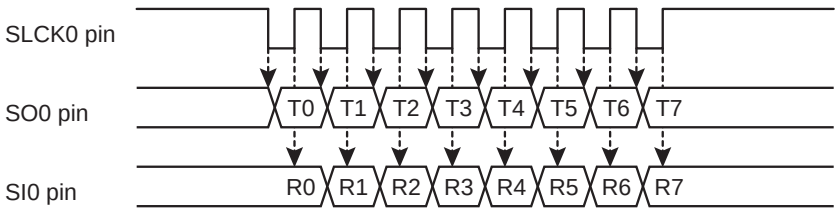
Table 19-4 Transfer Edge Selection

SIO0CR<SIOEDG>	Data transmission	Data reception
0	Falling edge	Rising edge
1	Rising edge	Falling edge

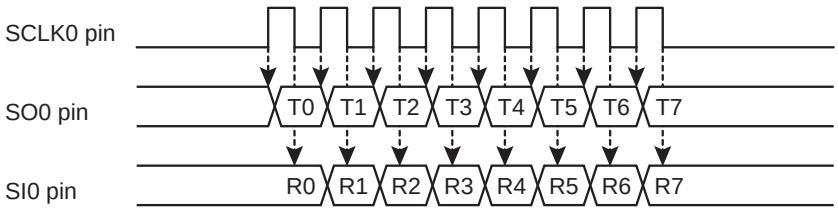
When SIOCR<SIOEDG> is "0", the data is transmitted in synchronization with the falling edge of the clock and the data is received in synchronization with the rising edge of the clock.

When SIOCR<SIOEDG> is "1", the data is transmitted in synchronization with the rising edge of the clock and the data is received in synchronization with the falling edge of the clock.

Downloaded from Arrow.com.



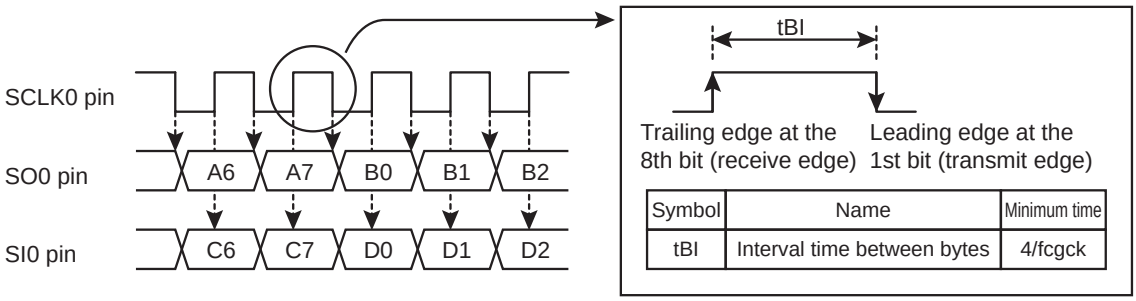
When SIOCR<SIOEDG>="0"



When SIOCR<SIOEDG>="1"

Figure 19-2 Transfer Edge

Note: When an external clock input is used, 4/fcgck or longer is needed between the receive edge at the 8th bit and the transfer edge at the first bit of the next transfer.



Symbol	Name	Minimum time
tBI	Interval time between bytes	4/fcgck

Figure 19-3 Interval time between bytes

Downloaded from Arrow.com.

19.5 Transfer Modes

19.5.1 8-bit transmit mode

The 8-bit transmit mode is selected by setting SIO0CR<SIOM> to "01".

19.5.1.1 Setting

Before starting the transmit operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit transmit mode is selected by setting SIO0CR<SIOM> to "01".

The transmit operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

19.5.1.2 Starting the transmit operation

The transmit operation is started by writing data to SIO0BUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and then transmitted as the serial data from the SO0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>. The serial data becomes undefined if the transmit operation is started without writing any transmit data to SIO0BUF.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1" and an INTSIO0 interrupt request is generated.

SIO0SR<SEF> is cleared to "0" when the 8th bit of the serial data is output.

19.5.1.3 Transmit buffer and shift operation

If data is written to SIO0BUF when the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0".

If data is written to SIO0BUF when some data remains in the shift register, SIO0SR<TBFL> is set to "1". If new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

19.5.1.4 Operation on completion of transmission

The operation on completion of the data transmission varies depending on the operating clock and the state of SIO0SR<TBFL>.

Downloaded from Arrow.com.

(1) When the internal clock is used and SIO0SR<TBFL> is "0"

When the data transmission is completed, the SCLK0 pin becomes the initial state and the SO0 pin becomes the "H" level. SIO0SR<SEF> remains at "0". When the internal clock is used, the serial clock and data output is stopped until the next transmit data is written into SIO0BUF (automatic wait).

When the subsequent data is written into SIO0BUF, SIO0SR<SEF> is set to "1", the SCLK0 pin outputs the serial clock, and the transmit operation is restarted. An INTSIO0 interrupt request is generated at the restart of the transmit operation.

(2) When an external clock is used and SIO0SR<TBFL> is "0"

When the data transmission is completed, the SO pin keeps last output value. When an external serial clock is input to the SCLK0 pin after completion of the data transmission, an undefined value is transmitted and the transmit underrun error flag SIO0SR<UERR> is set to "1".

If a transmit underrun error occurs, data must not be written to SIO0BUF during the transmission of an undefined value. (It is recommended to finish the transmit operation by setting SIO0CR<SIOS> to "0" or force the transmit operation to stop by setting SIO0CR<SIOM> to "00".)

The transmit underrun error flag SIO0SR<UERR> is cleared by reading SIO0SR.

(3) When an internal or external clock is used and SIO0SR<TBFL> is "1"

When the data transmission is completed, SIO0SR<TBFL> is cleared to "0". The data in SIO0BUF is transferred to the shift register and the transmission of subsequent data is started. At this time, SIO0SR<SEF> is set to "1" and an INTSIO0 interrupt request is generated.

19.5.1.5 Stopping the transmit operation

Set SIO0CR<SIOS> to "0" to stop the transmit operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the transmit operation is stopped immediately and an INTSIO0 interrupt request is generated. When SIO0SR<SEF> is "1", the transmit operation is stopped after all the data in the shift register is transmitted (reserved stop). At this time, an INTSIO0 interrupt request is generated again.

When the transmit operation is completed, SIO0SR<SIOF, SEF and TBFL> are cleared to "0". Other SIO0SR registers keep their values.

If the internal clock has been used, the SO0 pin automatically returns to the "H" level. If an external clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the "H" level, write "00" to SIO0CR<SIOM> when the operation is stopped.

The transmit operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. The SO0 pin becomes the "H" level. If the internal clock is selected, the SCLK0 pin returns to the initial level.

Downloaded from Arrow.com.

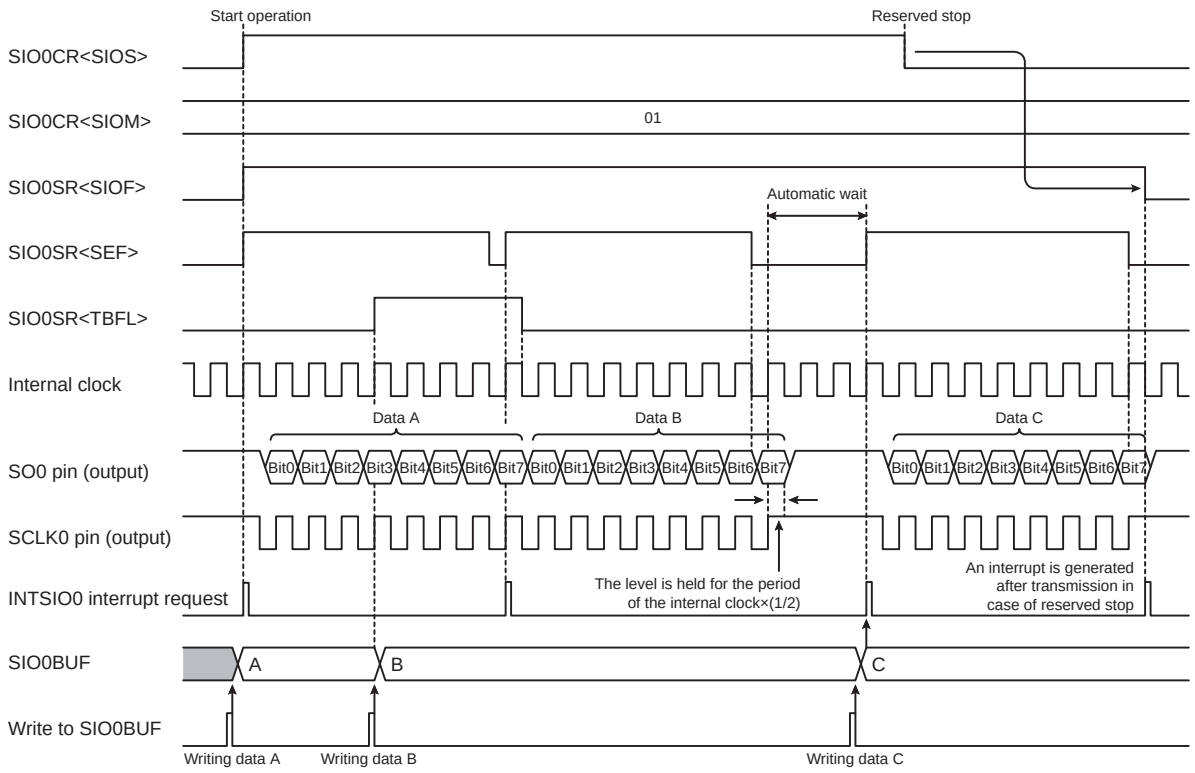


Figure 19-4 8-bit Transmit Mode (Internal Clock and Reserved Stop)

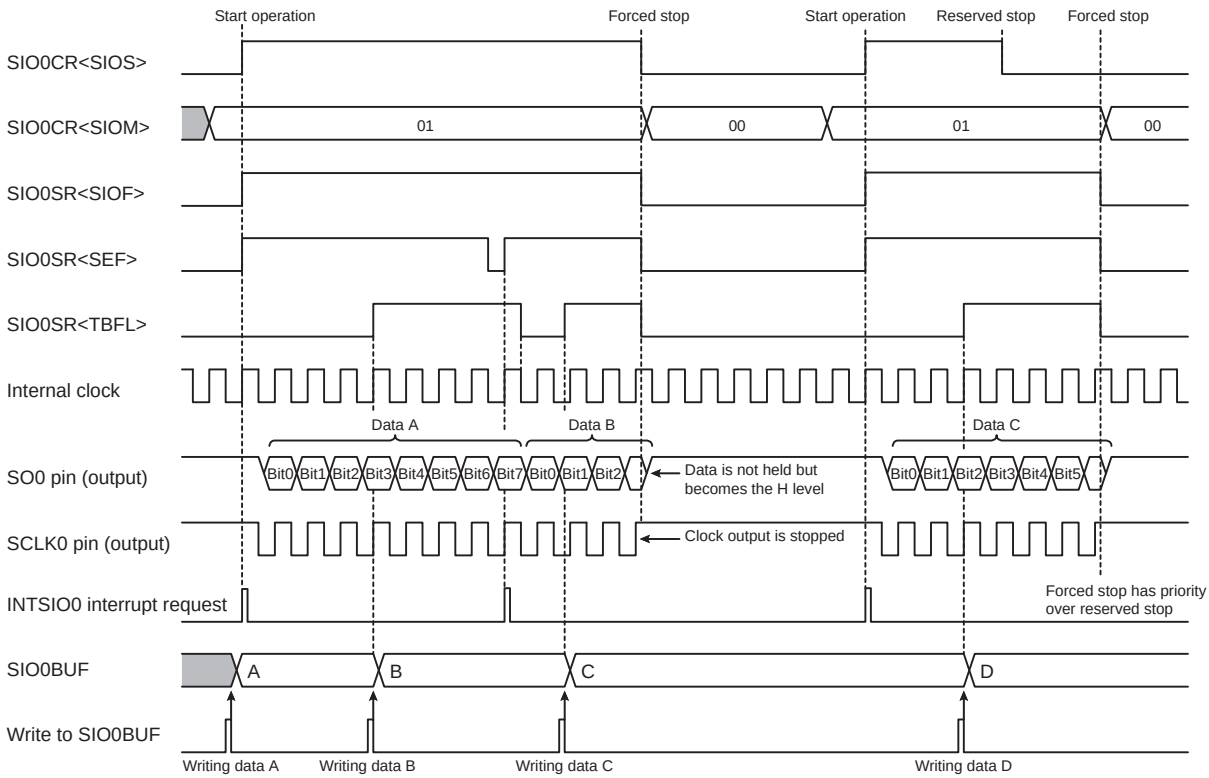


Figure 19-5 8-bit Transmit Mode (Internal Clock and Forced Stop)

Downloaded from Arrow.com.

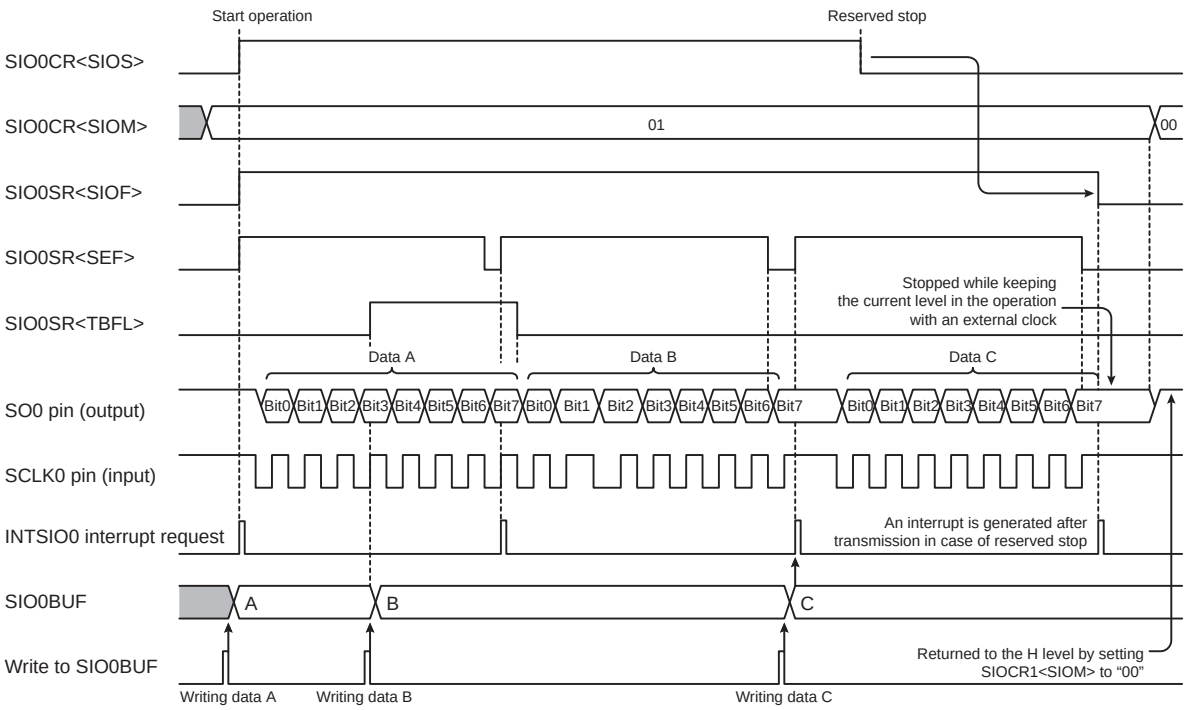


Figure 19-6 8-bit Transmit Mode (External Clock and Reserved Stop)

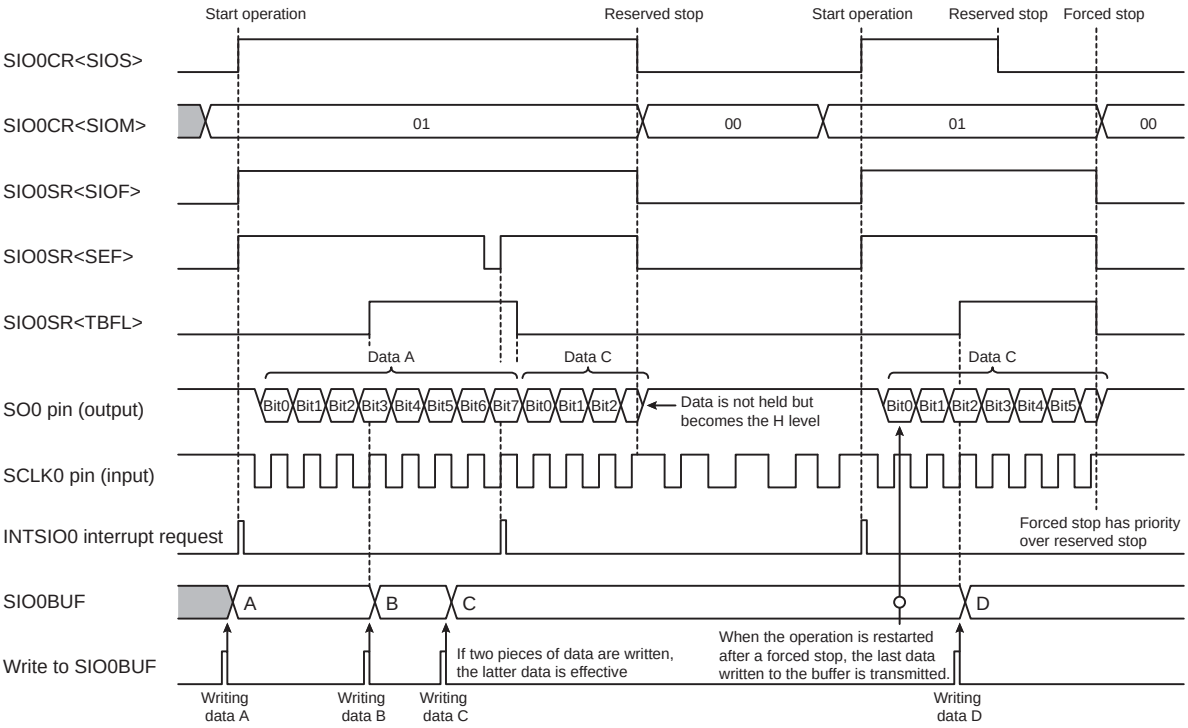


Figure 19-7 8-bit Transmit Mode (External Clock and Forced Stop)

Downloaded from Arrow.com.

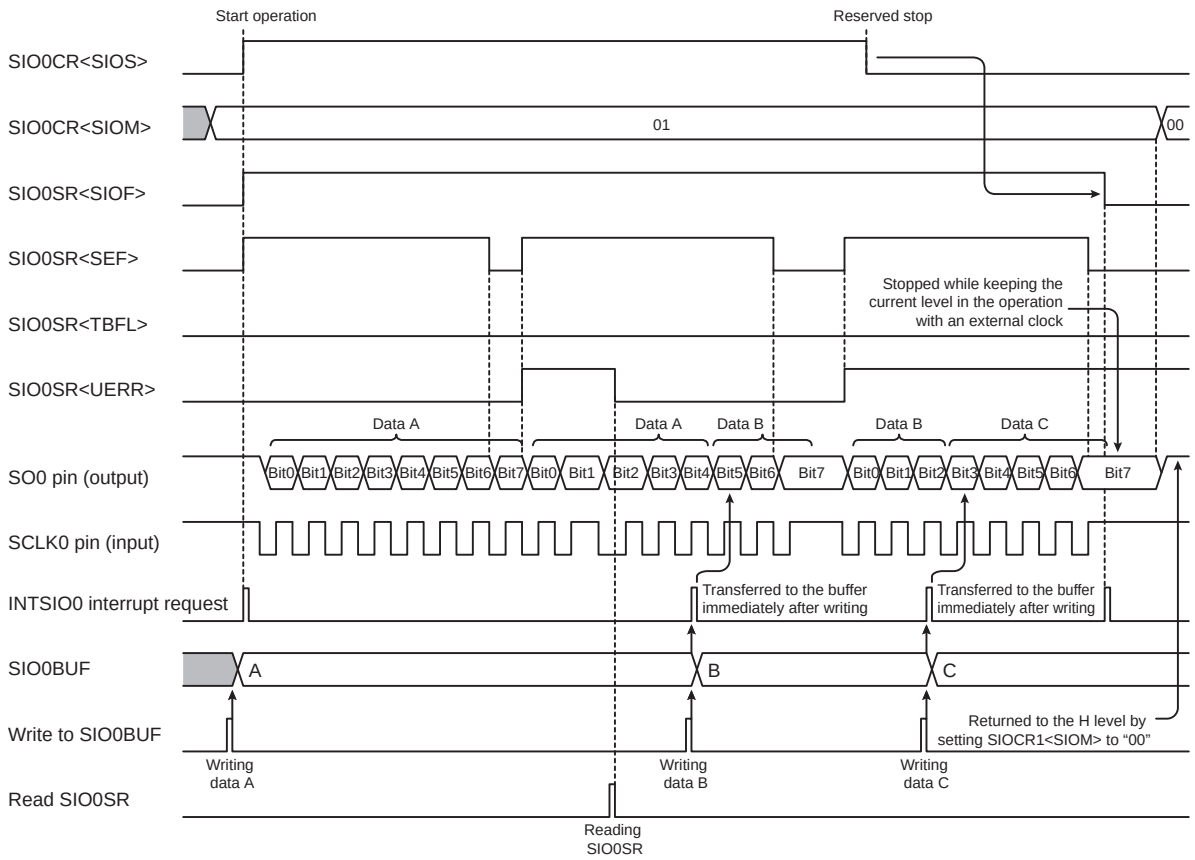


Figure 19-8 8-bit Transmit Mode (External Clock and Occurrence of Transmit Underrun Error)

Downloaded from Arrow.com.

19.5.2 8-bit Receive Mode

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to "10".

19.5.2.1 Setting

As in the case of the transmit mode, before starting the receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit receive mode is selected by setting SIO0CR<SIOM> to "10".

Reception is started by setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIO0CR<SIOS> is valid.

19.5.2.2 Starting the receive operation

Reception is started by setting SIO0CR<SIOS> to "1". External serial data is taken into the shift register from the SI0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1".

19.5.2.3 Operation on completion of reception

When the data reception is completed, the data is transferred from the shift register to SIO0BUF and an INTSIO0 interrupt request is generated. The receive completion flag SIO0SR<REND> is set to "1".

In the operation with the internal clock, the serial clock output is stopped until the receive data is read from SIO0BUF (automatic wait). At this time, SIO0SR<SEF> is set to "0". By reading the receive data from SIO0BUF, SIO0SR<SEF> is set to "1", the serial clock output is restarted and the receive operation continues.

In the operation with an external clock, data can be continuously received without reading the received data from SIO0BUF. In this case, data must be read from SIO0BUF before the subsequent data has been fully received. If the subsequent data is received completely before reading data from SIO0BUF, the overrun error flag SIO0SR<OERR> is set to "1". When an overrun error has occurred, set SIO0CR<SIOM> to "00" to abort the receive operation. The data received at the occurrence of an overrun error is discarded, and SIO0BUF holds the data value received before the occurrence of the overrun error.

SIO0SR<REND> is cleared to "0" by reading data from SIO0BUF. SIO0SR<OERR> is cleared by reading SIO0SR.

19.5.2.4 Stopping the receive operation

Set SIO0CR<SIOS> to "0" to stop the receive operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0 interrupt request is generated in this state.

Downloaded from Arrow.com.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data has been completely received (reserved stop). At this time, an INTSIO0 interrupt request is generated.

After the operation has stopped completely, SIO0SR<SIOF and SEF> are cleared to "0". Other SIO0SR registers keep their values.

The receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. If the internal clock is selected, the SCLK0 pin returns to the initial level.

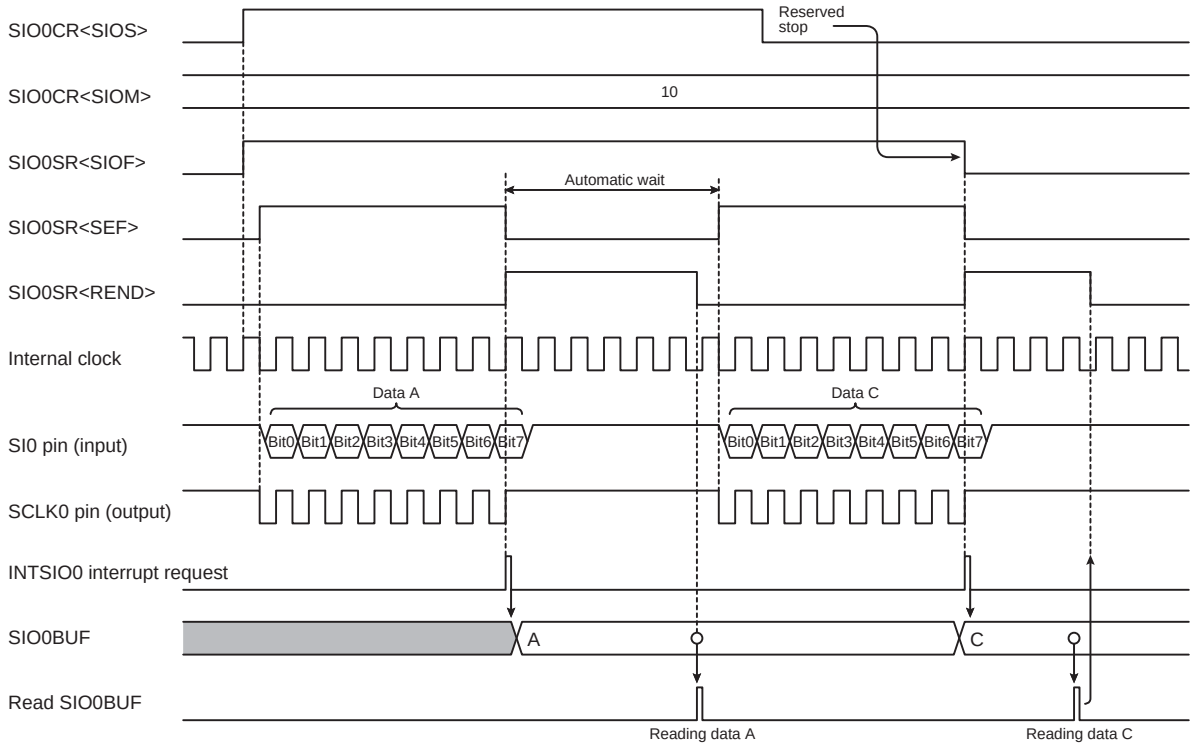


Figure 19-9 8-bit Receive Mode (Internal Clock and Reserved Stop)

Downloaded from Arrow.com.

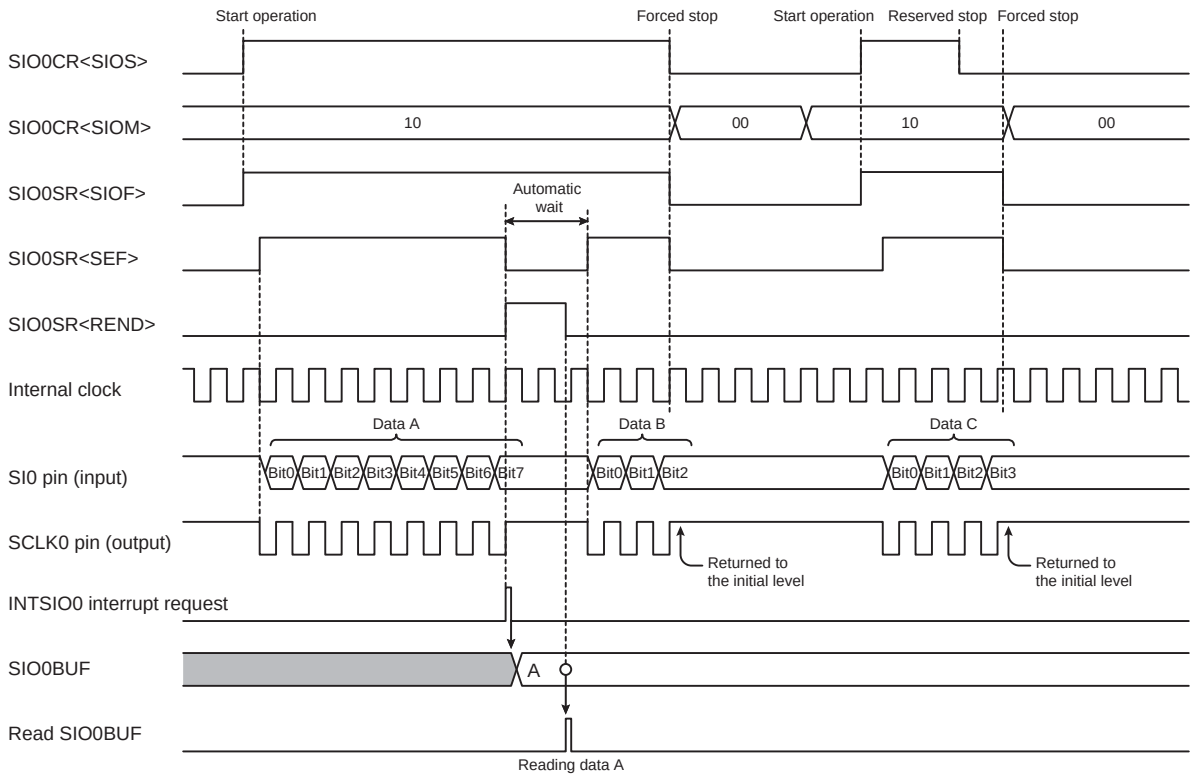


Figure 19-10 8-bit Receive Mode (Internal Clock and Forced Stop)

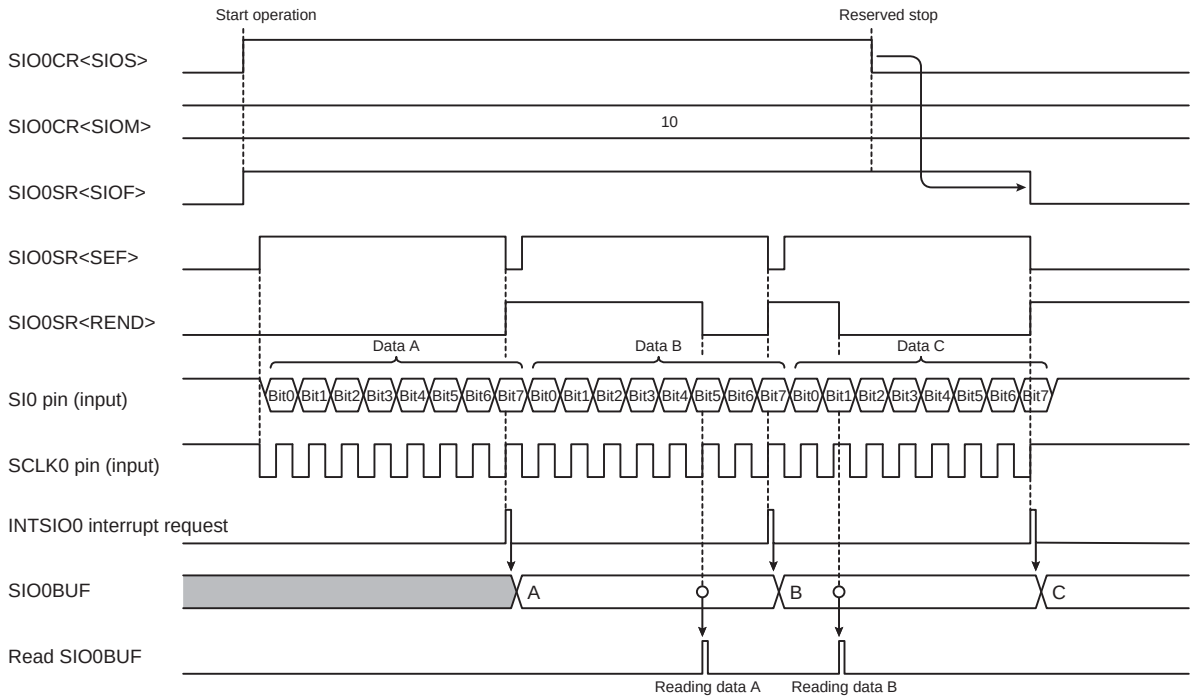


Figure 19-11 8-bit Receive Mode (External Clock and Reserved Stop)

Downloaded from Arrow.com.

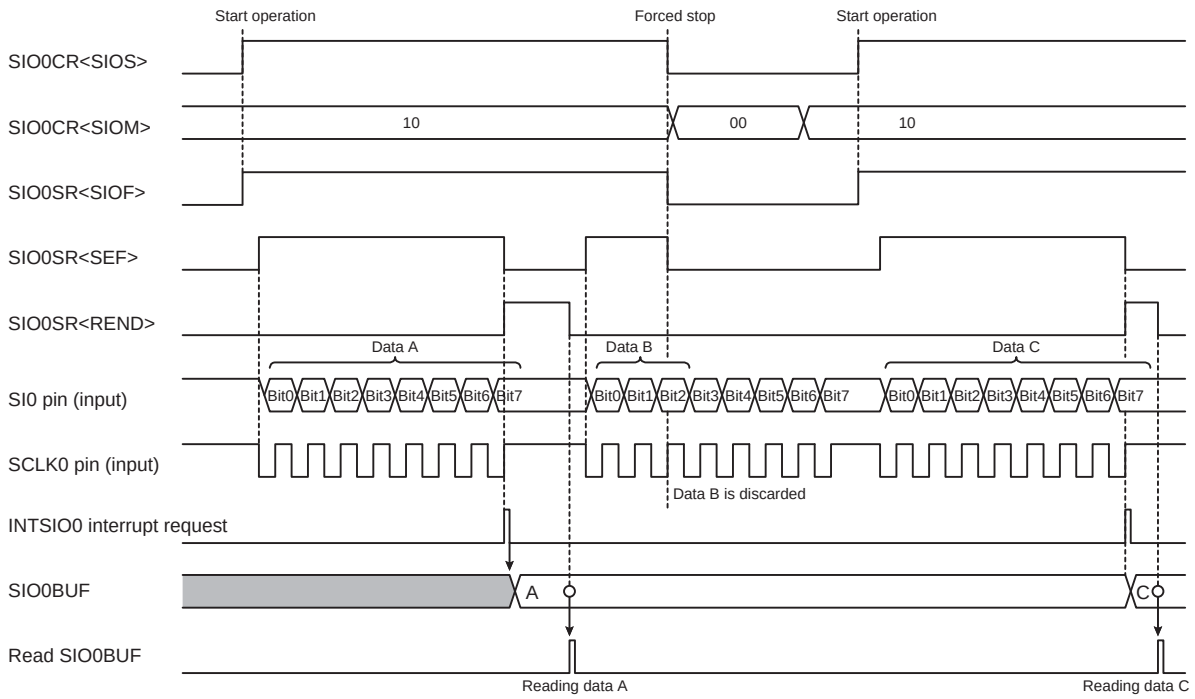


Figure 19-12 8-bit Receive Mode (External Clock and Forced Stop)

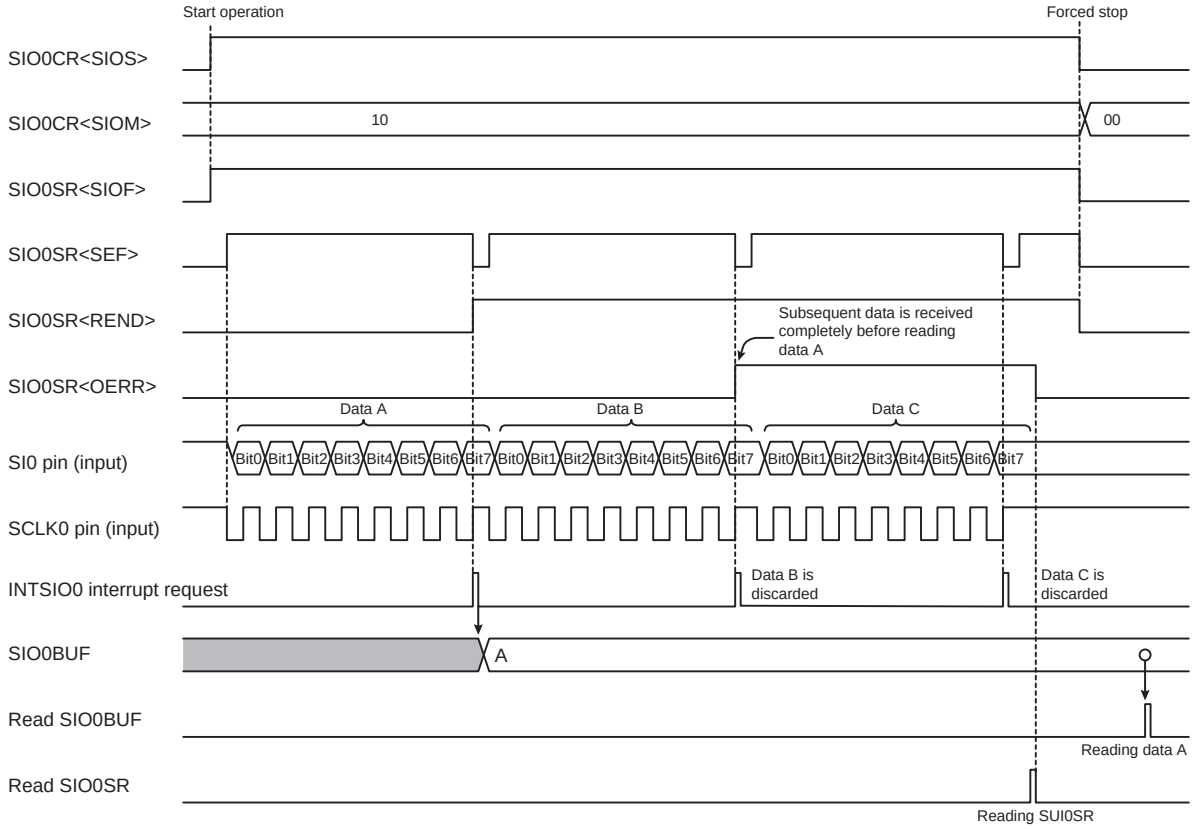


Figure 19-13 8-bit Receive Mode (External Clock and Occurrence of Overrun Error)

Downloaded from Arrow.com.

19.5.3 8-bit transmit/receive mode

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

19.5.3.1 Setting

Before starting the transmit/receive operation, select the transfer edges at SIO0CR<SIOEDG>, a transfer format at SIO0CR<SIODIR> and a serial clock at SIO0CR<SIOCKS>. To use the internal clock as the serial clock, select an appropriate serial clock at SIO0CR<SIOCKS>. To use an external clock as the serial clock, set SIO0CR<SIOCKS> to "111".

The 8-bit transmit/receive mode is selected by setting SIO0CR<SIOM> to "11".

The transmit/receive operation is started by writing the first byte of transmit data to SIO0BUF and then setting SIO0CR<SIOS> to "1".

Writing data to SIO0CR<SIOEDG, SIOCKS and SIODIR> is invalid when the serial communication is in progress, or when SIO0SR<SIOF> is "1". Make these settings while the serial communication is stopped. While the serial communication is in progress (SIO0SR<SIOF>="1"), only writing "00" to SIO0CR<SIOM> or writing "0" to SIOCR<SIOS> is valid.

19.5.3.2 Starting the transmit/receive operation

The transmit/receive operation is started by writing data to SIO0BUF and then setting SIO0CR<SIOS> to "1". The transmit data is transferred from SIO0BUF to the shift register, and the serial data is transmitted from the SO0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>. At the same time, the serial data is received from the SI0 pin according to the settings of SIO0CR<SIOEDG, SIOCKS and SIODIR>.

In the internal clock operation, the serial clock of the selected baud rate is output from the SCLK0 pin. In the external clock operation, an external clock must be supplied to the SCLK0 pin.

The transmit data becomes undefined if the transmit/receive operation is started without writing any transmit data to SIO0BUF.

By setting SIO0CR<SIOS> to "1", SIO0SR<SIOF and SEF> are automatically set to "1" and an INTSIO0 interrupt request is generated.

SIO0SR<SEF> is cleared to "0" when the 8th bit of data is received.

19.5.3.3 Transmit buffer and shift operation

If any data is written to SIO0BUF when the serial communication is in progress and the shift register is empty, the written data is transferred to the shift register immediately. At this time, SIO0SR<TBFL> remains at "0".

If any data is written to SIO0BUF when some data remains in the shift register, SIO0SR<TBFL> is set to "1". If new data is written to SIO0BUF in this state, the contents of SIO0BUF are overwritten by the new value. Make sure that SIO0SR<TBFL> is "0" before writing data to SIO0BUF.

19.5.3.4 Operation on completion of transmission/reception

When the data transmission/reception is completed, SIO0SR<REND> is set to "1" and an INTSIO0 interrupt request is generated. The operation varies depending on the operating clock.

Downloaded from Arrow.com.

(1) When the internal clock is used

If SIO0SR<TBFL> is "1", it is cleared to "0" and the transmit/receive operation continues. If SIO0SR<REND> is already "1", SIO0SR<OERR> is set to "1".

If SIO0SR<TBFL> is "0", the transmit/receive operation is aborted. The SCLK0 pin becomes the initial state and the SO0 pin becomes the "H" level. SIO0SR<SEF> remains at "0". When the subsequent data is written to SIO0BUF, SIO0SR<SEF> is set to "1", the SCLK0 pin outputs the clock and the transmit/receive operation is restarted. To confirm the receive data, read it from SIO0BUF before writing data to SIO0BUF.

(2) When an external clock is used

The transmit/receive operation continues. If the external serial clock is input without writing any data to SIO0BUF, the last data value set to SIO0BUF is re-transmitted. At this time, the transmit underrun error flag SIO0SR<UERR> is set to "1".

When the next 8-bit data is received completely before SIO0BUF is read, or in the state of SIO0SR<REND>="1", SIO0SR<OERR> is set to "1".

19.5.3.5 Stopping the transmit/receive operation

Set SIO0CR<SIOS> to "0" to stop the transmit/receive operation. When SIO0SR<SEF> is "0", or when the shift operation is not in progress, the operation is stopped immediately. Unlike the transmit mode, no INTSIO0 interrupt request is generated in this state.

When SIO0SR<SEF> is "1", the operation is stopped after the 8-bit data is received completely. At this time, an INTSIO0 interrupt request is generated.

After the operation has stopped completely, SIO0SR<SIOF, SEF and TBFL> are cleared to "0". Other SIO0SR registers keep their values.

If the internal clock has been used, the SO0 pin automatically returns to the "H" level. If an external clock has been used, the SO0 pin keeps the last output value. To return the SO0 pin to the "H" level, write "00" to SIO0CR<SIOM> when the operation is stopped.

The transmit/receive operation can be forced to stop by setting SIO0CR<SIOM> to "00" during the operation. By setting SIO0CR<SIOM> to "00", SIO0CR<SIOS> and SIO0SR are cleared to "0" and the SIO stops the operation, regardless of the SIO0SR<SEF> value. The SO0 pin becomes the "H" level. If the internal clock is selected, the SCLK0 pin returns to the initial level.

Downloaded from Arrow.com.

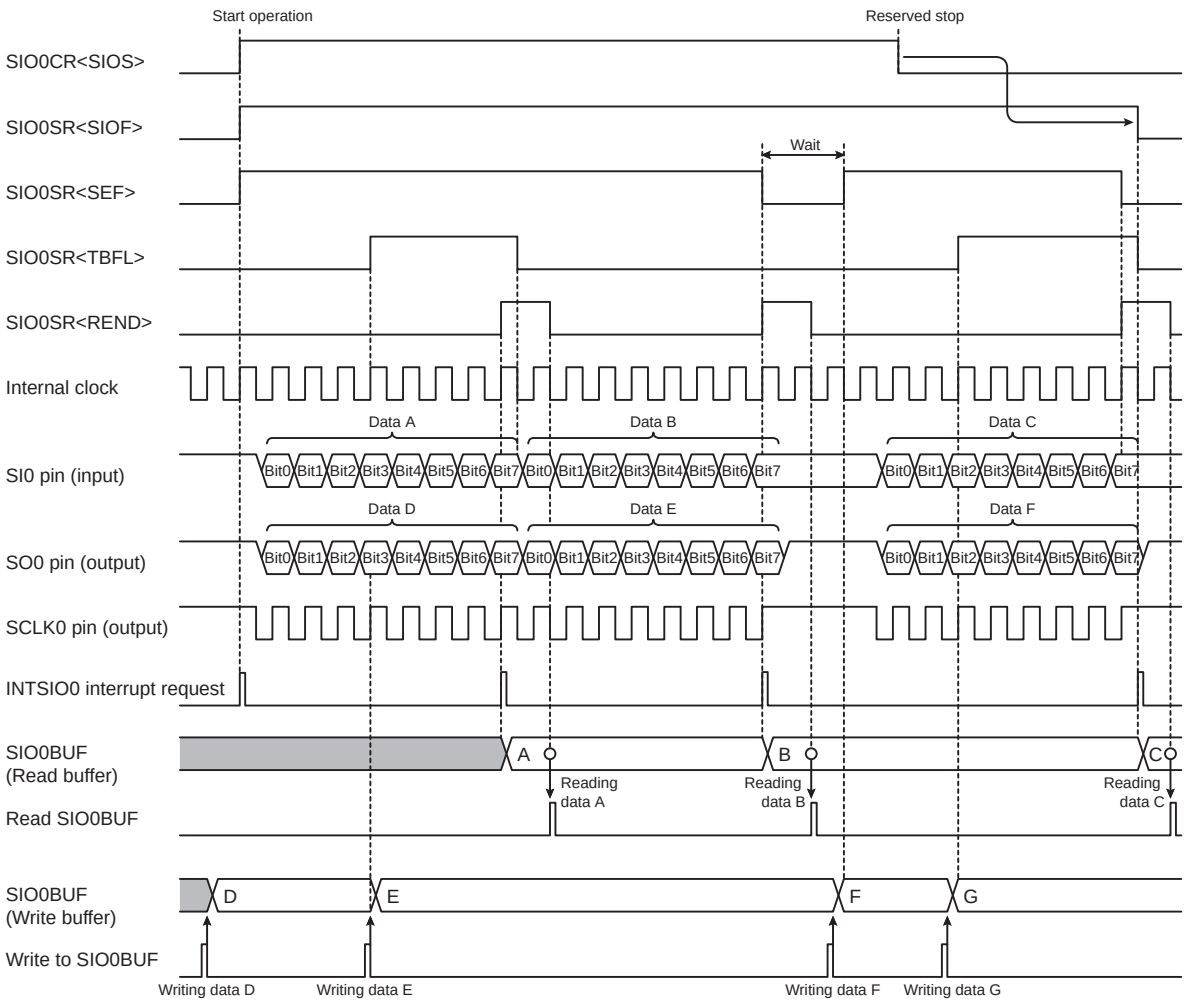


Figure 19-14 8-bit Transmit/Receive Mode (Internal Clock and Reserved Stop)

Downloaded from Arrow.com.

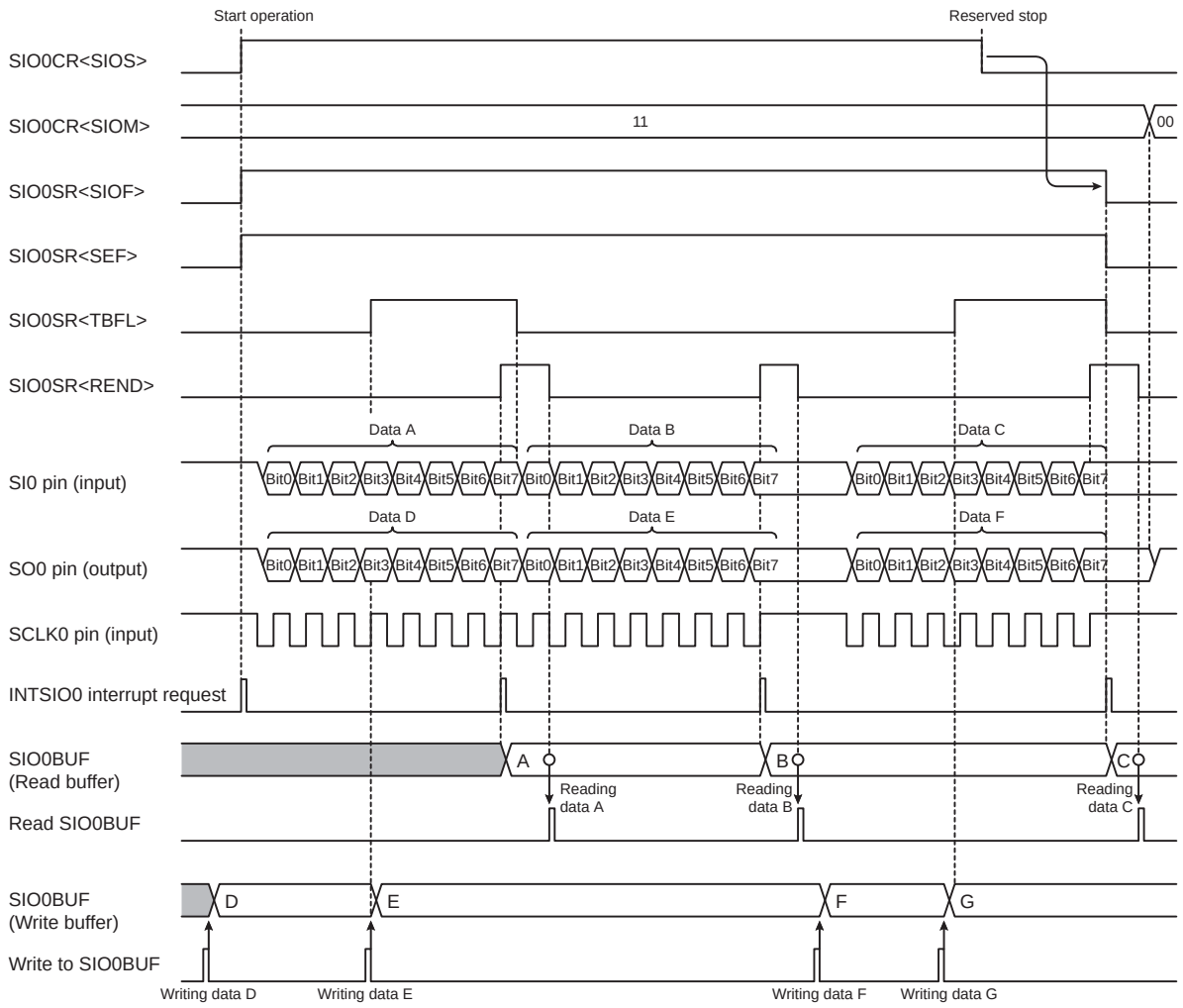


Figure 19-15 8-bit Transmit/Receive Mode (External Clock and Reserved Stop)

Downloaded from Arrow.com.

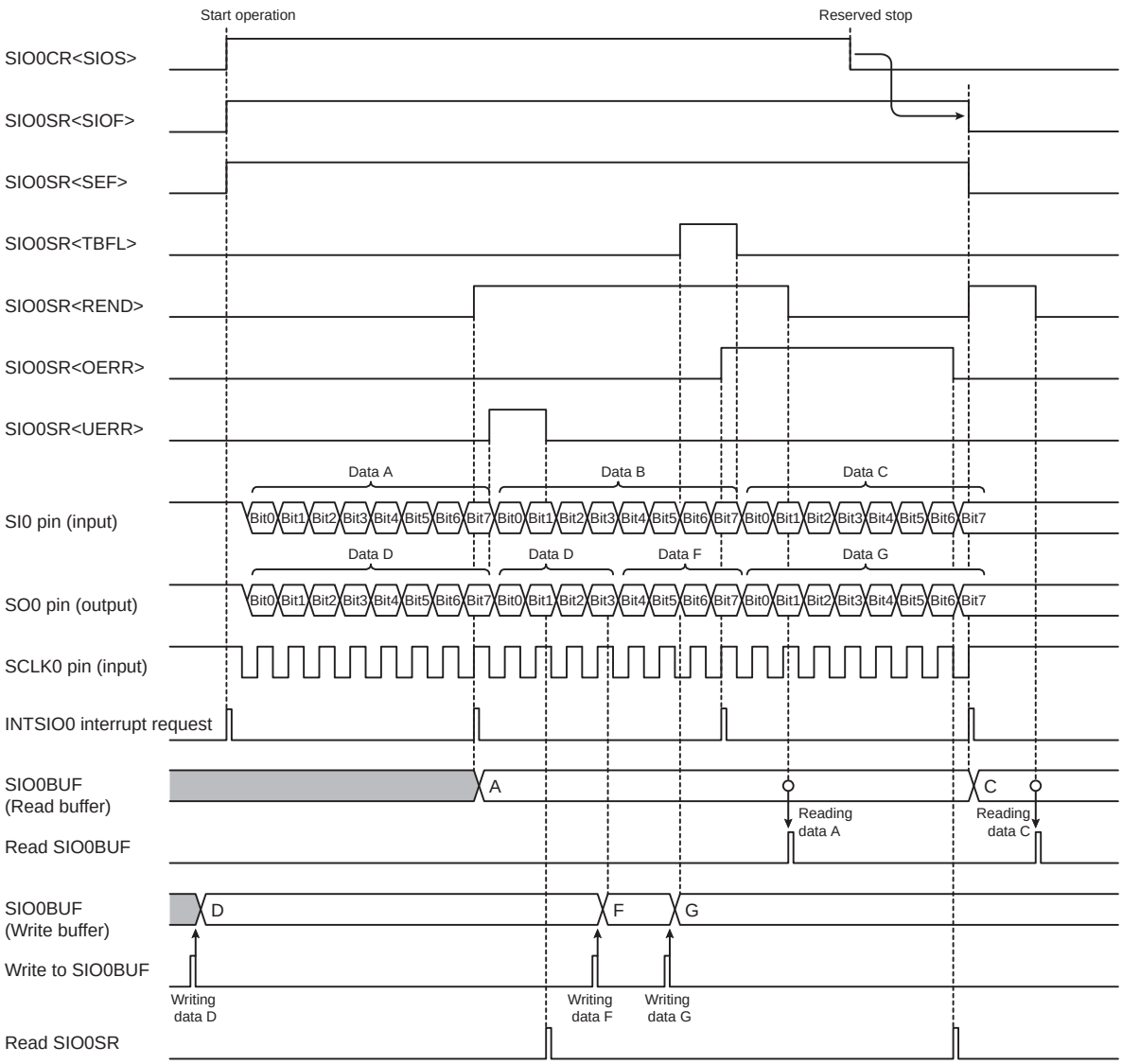


Figure 19-16 8-bit Transmit/Receive Mode (External Clock, Occurrence of Transmit Under-run Error and Occurrence of Overrun Error)

Downloaded from Arrow.com.

19.6 AC Characteristics

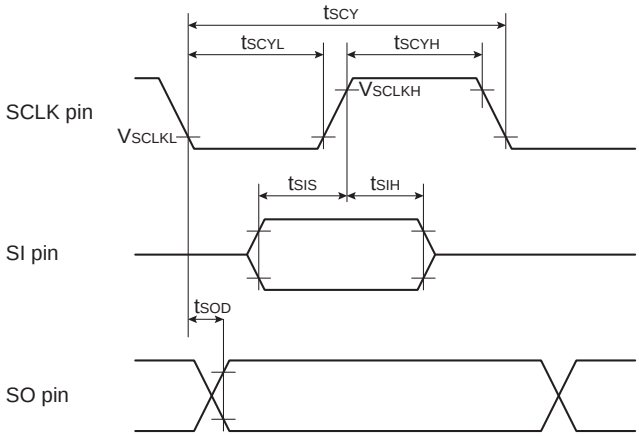


Figure 19-17 AC Characteristics

(V_{SS} = 0 V, V_{DD} = 4.5 V - 5.5 V, Topr = -40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
SCLK cycle time	t_{SCY}	Internal clock operation SO pin and SCLK pin load capacity=100 pF	2 / fcgck	-	-	ns
SCLK "L" pulse width	t_{SCYL}		1 / fcgck − 25	-	-	
SCLK "H" pulse width	t_{SCYH}		1 / fcgck − 15	-	-	
SI input setup time	t_{SIS}		60	-	-	
SI input hold time	t_{SIH}		35	-	-	
SO output delay time	t_{SOD}		−50	-	50	
SCLK cycle time	t_{SCY}	External clock operation SO pin and SCLK pin load capacity=100 pF	2 / fcgck	-	-	
SCLK "L" pulse width	t_{SCYL}		1 / fcgck	-	-	
SCLK "H" pulse width	t_{SCYH}		1 / fcgck	-	-	
SI input setup time	t_{SIS}		50	-	-	
SI input hold time	t_{SIH}		50	-	-	
SO output delay time	t_{SOD}		0	-	60	
SCLK low-level input voltage	t_{SCLKL}		0	-	V_{DD} × 0.30	V
SCLK high-level input voltage	t_{SCLKH}		V_{DD} × 0.70	-	V_{DD}	

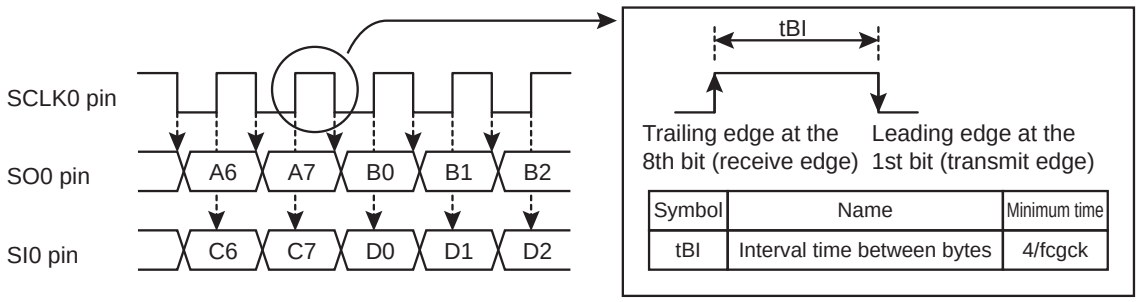


Figure 19-18 Interval time between bytes

Downloaded from Arrow.com.

Downloaded from Arrow.com.

20. Serial Bus Interface (SBI)

The TMP89FW24A contains 1 channels of serial bus interface (SBI).

The serial bus interface supports serial communication conforming to the I^2C bus standards. It has clock synchronization and arbitration functions, and supports the multi-master in which multiple masters are connected on a bus. It also supports the unique free data format.

Downloaded from Arrow.com.

20.1 Communication Format

20.1.1 I²C bus

The I²C bus is connected to devices via the SDA0 and SCL0 pins and can communicate with multiple devices.

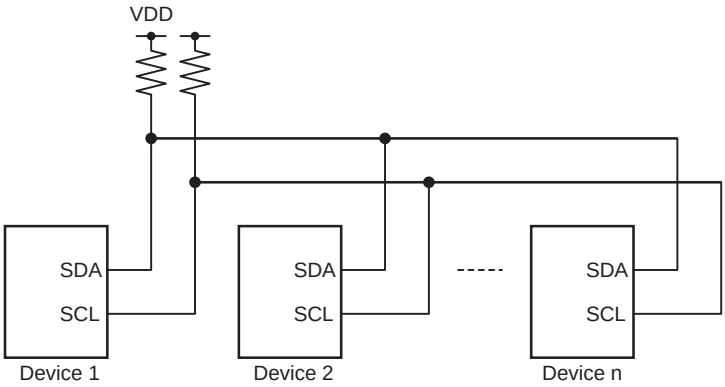


Figure 20-1 Device Connections

Communications are implemented between a master and slave.

The master transmits the start condition, the slave addresses, the direction bit and the stop condition to the slave(s) connected to the bus, and transmits and receives data.

The slave detects these conditions transmitted from the master by the hardware, and transmits and receives data.

The data format of the I²C bus that can communicate via the serial bus interface is shown in Figure 20-2.

The serial bus interface does not support the following functions among those specified by the I²C bus standards:

1. Start byte
2. 10-bit addressing
3. SDA and SCL pins falling edge slope control

Downloaded from Arrow.com.

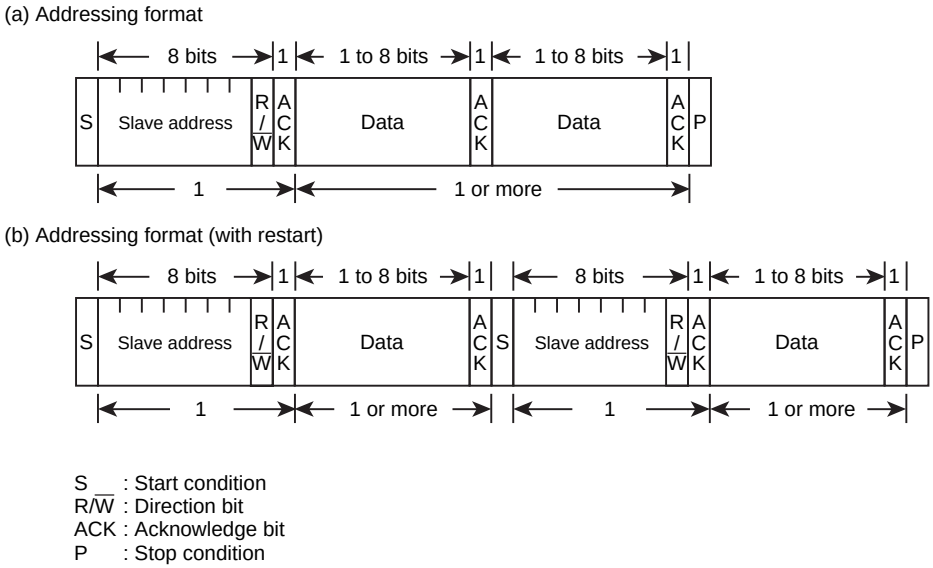


Figure 20-2 Data Format of I²C Bus

20.1.2 Free data format

The free data format is for communication between a master and slave.

In the free data format, the slave address and the direction bit are processed as data.

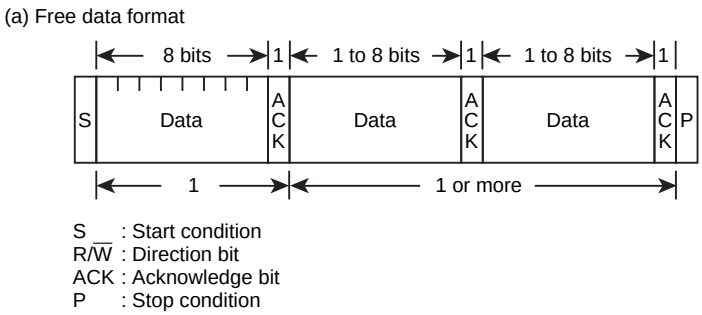


Figure 20-3 Free Data Format

Downloaded from Arrow.com.

20.2 Configuration

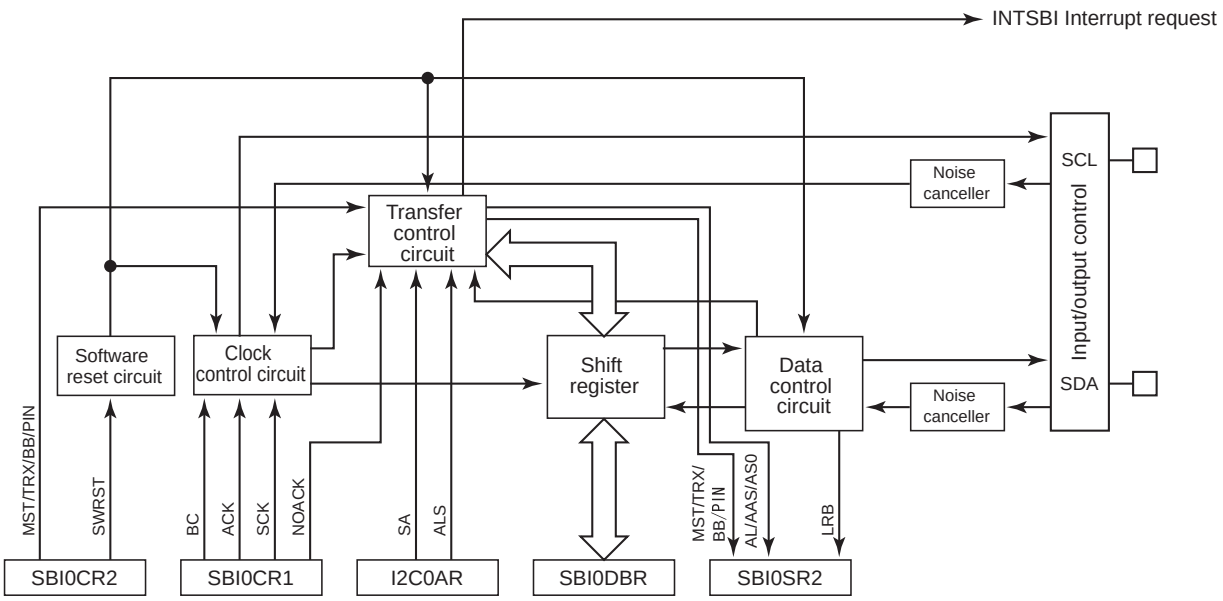


Figure 20-4 Serial Bus Interface 0 (SBI0)

Downloaded from Arrow.com.

20.3 Control

The following registers are used to control the serial bus interface and monitor the operation status.

- Serial bus interface control register 1 (SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface status register 2 (SBI0SR2)
- Serial bus interface data buffer register (SBI0DBR)
- I^2C bus address register (I2C0AR)

In addition, the serial bus interface has low power consumption registers that save power when the serial bus interface is not being used.

Low power consumption register 1

POFFCR1		7	6	5	4	3	2	1	0
(0x00F75)	Bit Symbol	-	-	-	SBI0EN	-	UART2EN	UART1EN	UART0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

SBI0EN	I2C0 Control	0 1	Disable Enable
UART2EN	UART2 Control	0 1	Disable Enable
UART1EN	UART1 Control	0 1	Disable Enable
UART0EN	UART0 Control	0 1	Disable Enable

Note 1: When SBI0EN is cleared to "0", the clock supply to the serial bus interface is stopped. At this time, the data written to the serial bus interface control registers is invalid. When the serial bus interface is used, set SBI0EN to "1" and then write the data to the serial bus interface control registers.

Downloaded from Arrow.com.

Serial bus interface control register 1

SBI0CR1 (0x00022)

	7	6	5	4	3	2	1	0
Bit Symbol	BC			ACK	NOACK	SCK		
Read/Write	R/W			R/W	R/W	R/W		
After reset	0	0	0	0	0	0	0	0

Note 1: fcgck: Gear clock [Hz], fs: Low-frequency clock oscillation circuit clock

Note 2: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: After a software reset is generated, all the bits of SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Note 5: When fcgck is 4MHz, SCK should be not set to 0y000, 0y001 or 0y010 because it is not possible to satisfy the bus specification of fast mode.

Serial bus interface control register 2

SBI0CR2 (0x00023)

	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	SBIM	-	SWRST	
Read/Write	W	W	W	W	W	R	W	
After reset	0	0	0	1	0	0	0	

Note 1: When SBI0CR2<SBIM> is "0", no value can be written to SBI0CR2 except SBI0CR2<SBIM>. Before writing values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

Note 2: Don't change the contents of the registers, except SBI0CR2<SWRST>, when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: Make sure that the port is in a high state before switching the port mode to the serial bus interface mode. Make sure that the bus is free before switching the serial bus interface mode to the port mode.

Note 4: SBI0CR2 is a write-only register, and must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 5: After a software reset is generated, all the bits of SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 6: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Serial bus interface status register 2

SBI0SR2 (0x00023)

	7	6	5	4	3	2	1	0
Bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R	R	R	R	R	R	R	R
After reset	0	0	0	1	0	0	0	*

Note 1: * : Unstable

Note 2: When SBI0CR2<SBIM> becomes "0", SBI0SR is initialized.

Note 3: After a software reset is generated, all the bits of the SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Downloaded from Arrow.com.

I²C bus address register

I2C0AR (0x00024)

	7	6	5	4	3	2	1	0
Bit Symbol	SA							ALS
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0

Note 1: Don't set I2C0AR<SA> to "0x00". If it is set to "0x00", the slave address is deemed to be matched when the I²C bus standard start byte ("0x01") is received in the slave mode.

Note 2: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 3: After a software reset is generated, all the bits of the SBI0CR2 register except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR and SBI0SR2 registers are initialized.

Note 4: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Serial bus interface data buffer register

SBI0DBR (0x00025)

	7	6	5	4	3	2	1	0
Bit Symbol	SBI0DBR							
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0

Note 1: Write the transmit data beginning with the most significant bit (bit 7).

Note 2: SBI0DBR has individual writing and reading buffers, and written data cannot be read out. Therefore, SBI0DBR must not be accessed by using a read-modify-write instruction, such as a bit operation.

Note 3: Don't change the contents of the registers when the start condition is generated, the stop condition is generated or the data transfer is in progress. Write data to the registers before the start condition is generated or during the period from when an interrupt request is generated for stopping the data transfer until it is released.

Note 4: To set SBI0CR2<PIN> to "1" by writing the dummy data to SBI0DBR, write 0x00. Writing any data other than 0x00 causes an improper value in the subsequently received data.

Note 5: When the operation is switched to STOP, IDLE0 or SLOW mode, the SBI0CR2 register, except SBI0CR2<SBIM>, and the SBI0CR1, I2C0AR and SBI0DBR registers are initialized.

Downloaded from Arrow.com.

20.4 Functions

20.4.1 Low Power Consumption Function

The serial bus interface has a low power consumption register (POFFCR1) that saves power when the serial bus interface is not being used.

Setting POFFCR1<SBI0EN> to "0" disables the basic clock supply to the serial bus interface to save power. Note that this makes the serial bus interface unusable. Setting POFFCR1<SBI0EN> to "1" enables the basic clock supply to the serial bus interface and makes external interrupts usable.

After reset, POFFCR1<SBI0EN> is initialized to "0", and this makes the serial bus interface unusable. When using the serial bus interface for the first time, be sure to set POFFCR1<SBI0EN> to "1" in the initial setting of the program (before the serial bus interface control registers are operated).

Do not change POFFCR1<SBI0EN> to "0" during the serial bus interface operation, otherwise serial bus interface may operate unexpectedly.

20.4.2 Selecting the slave address match detection and the GENERAL CALL detection

SBI0CR1<NOACK> enables and disables the slave address match detection and the GENERAL CALL detection in the slave mode.

Clearing SBI0CR1<NOACK> to "0" enables the slave address match detection and the GENERAL CALL detection.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. The slave addresses and "GENERAL CALL" sent from the master are ignored. No acknowledgement is returned and no interrupt request is generated.

In the master mode, SBI0CR1<NOACK> is ignored and has no influence on the operation.

Note: If SBI0CR1<NOACK> is cleared to "0" during data transfer in the slave mode, it remains at "1" and returns an acknowledge signal of data transfer.

20.4.3 Selecting the number of clocks for data transfer and selecting the acknowledgement or non-acknowledgment mode

1-word data transfer consists of data and an acknowledge signal. When the data transfer is finished, an interrupt request is generated.

SBI0CR1<BC> is used to select the number of bits of data to be transmitted/received subsequently.

The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

The master device generates the clocks for an acknowledge signal and outputs an acknowledge signal in the receiver mode. The slave device counts the clocks for an acknowledge signal and outputs an acknowledge signal in the receiver mode.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".

The master device does not generate the clocks for an acknowledge signal. The slave device does not count the clocks for an acknowledge signal.

20.4.3.1 Number of clocks for data transfer

The number of clocks for data transfer is set by using SBI0CR1<BC> and SBI0CR1<ACK>.

The acknowledgment mode is activated by setting SBI0CR1<ACK> to "1".

Downloaded from Arrow.com.

In the acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, generates the clocks for an acknowledge signal, and generates an interrupt request.

The slave device counts the clocks that correspond to the data bits, counts the clocks for an acknowledge signal, and generates an interrupt request.

The non-acknowledgment mode is activated by setting SBI0CR1<ACK> to "0".

In the non-acknowledgment mode, the master device generates the clocks that correspond to the number of data bits, and generates an interrupt request.

The slave device counts the clocks that correspond to the data bits, and generates an interrupt request.

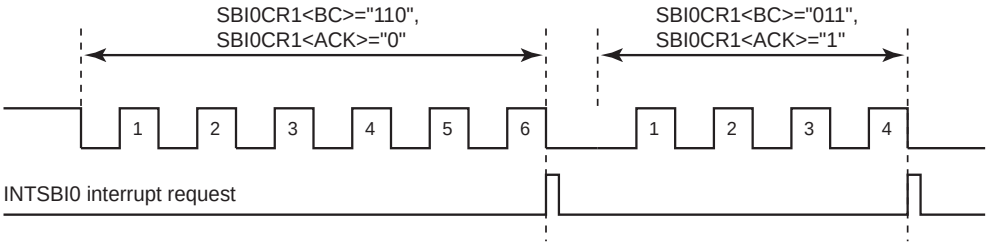


Figure 20-5 Number of Clocks for Data Transfer and SBI0CR1<BC> and SBI0CR1<ACK>

The relationship between the number of clocks for data transfer and SBI0CR1<BC> and SBI0CR1<ACK> is shown in Table 20-1.

Table 20-1 Relationship between the Number of Clocks for Data Transfer and SBI0CR1<BC> and SBI0CR1<ACK>

BC	ACK=0 (Non-acknowledgment mode)		ACK=1 (Acknowledgment mode)	
	Number of clocks for data transfer	Number of data bits	Number of clocks for data transfer	Number of data bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

BC is cleared to "000" by the start condition.

Therefore, the slave address and the direction bit are always transferred in 8-bit units. In other cases, BC keeps the set value.

Note: SBI0CR1<ACK> must be set before transmitting or receiving a slave address. When SBI0CR1<ACK> is cleared, the slave address match detection and the direction bit detection are not executed properly.

20.4.3.2 Output of an acknowledge signal

In the acknowledgment mode, the SDA0 pin changes as follows during the period of the clocks for an acknowledge signal.

- In the master mode

Downloaded from Arrow.com.

In the transmitter mode, the SDA0 pin is released to receive an acknowledge signal from the receiver during the period of the clocks for an acknowledge signal. In the receiver mode, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated during the period of the clocks for an acknowledge signal.

- In the slave mode

When a match between the received slave address and the slave address set to I2C0AR<SA> is detected or when a GENERAL CALL is received, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated during the period of the clocks for an acknowledge signal.

During the data transfer after the slave address match is detected or a "GENERAL CALL" is received in the transmitter mode, the SDA0 pin is released to receive an acknowledge signal from the receiver during the period of the clocks for an acknowledge signal.

In the receiver mode, the SDA0 pin is pulled down to the low level and an acknowledge signal is generated. Table 20-2 shows the states of the SCL0 and SDA0 pins in the acknowledgment mode.

Note: In the non-acknowledgment mode, the clocks for an acknowledge signal are not generated or counted, and thus no acknowledge signal is output.

Table 20-2 States of the SCL0 and SDA0 Pins in the Acknowledgment Mode

Mode	Pin	Condition	Transmitter	Receiver
Master	SCL0	-	Add the clocks for an acknowledge signal.	Add the clocks for an acknowledge signal
	SDA0	-	Release the pin to receive an acknowledge signal	Output the low level as an acknowledge signal to the pin
Slave	SCL0	-	Count the clocks for an acknowledge signal	Count the clocks for an acknowledge signal
	SDA0	When the slave address match is detected or a "GENERAL CALL" is received	-	Output the low level as an acknowledge signal to the pin
		During transfer after the slave address match is detected or a "GENERAL CALL" is received	Release the pin to receive an acknowledge signal	Output the low level as an acknowledge signal to the pin

20.4.4 Serial clock

20.4.4.1 Clock source

SBI0CR1<SCK> is used to set the HIGH and LOW periods of the serial clock to be output in the master mode.

SCK	t_{HIGH}(m/fcgck)	t_{LOW}(n/fcgck)
	m	n
000:	9	12
001:	11	14
010:	15	18
011:	23	26
100:	39	42
101:	71	74
110:	135	138
111:	263	266

Downloaded from Arrow.com.

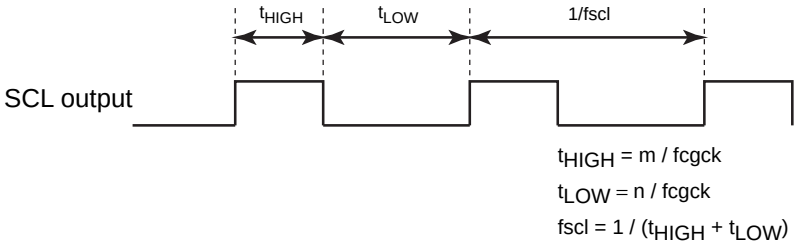


Figure 20-6 SCL Output

Note: There are cases where the HIGH period differs from t_{HIGH} selected at SBI0CR1<SCK> when the rising edge of the SCL pin becomes blunt due to the load capacity of the bus.

In the master mode, the hold time when the start condition is generated is t_{HIGH} [s] and the setup time when the stop condition is generated is t_{HIGH} [s].

When SBI0CR2<PIN> is set to "1" in the slave mode, the time that elapses before the release of the SCL pin is t_{LOW} [s].

In both the master and slave modes, the high level period must be 3/fcgck[s] or longer and the low level period must be 5/fcgck[s] or longer for the externally input clock, regardless of the SBI0CR1<SCK> setting.

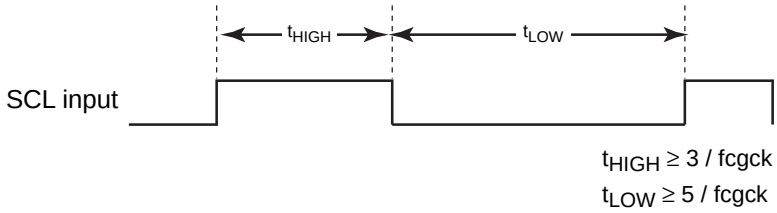


Figure 20-7 SCL Input

20.4.4.2 Clock synchronization

In the I²C bus, due to the structure of the pin, in order to drive a bus with a wired AND, a master device which pulls down a clock pulse to low will, in the first place, invalidate the clock pulse of another master device which generates a high-level clock pulse. Therefore, the master outputting the high level must detect this to correspond to it.

The serial bus interface circuit has a clock synchronization function. This function ensures normal transfer even if there are two or more masters on the same bus.

The example explains clock synchronization procedures when two masters simultaneously exist on a bus.

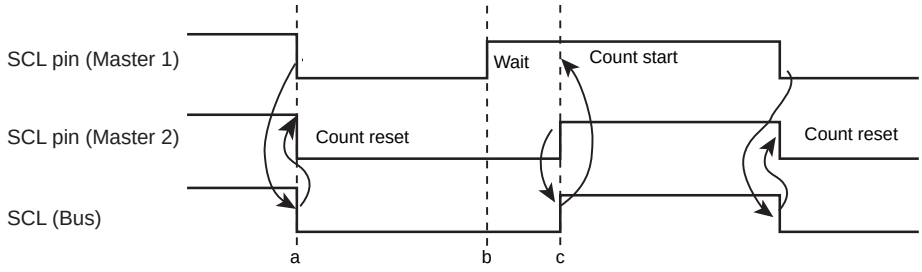


Figure 20-8 Example of Clock Synchronization

Downloaded from Arrow.com.

As Master 1 pulls down the SCL pin to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, Master 2 resets counting a clock pulse in the high level and sets the SCL pin to the low level.

Master 1 finishes counting a clock pulse in the low level at point "b" and sets the SCL pin to the high level. Since Master 2 holds the SCL line of the bus at the low level, Master 1 waits for counting a clock pulse in the high level. After Master 2 sets a clock pulse to the high level at point "c" and detects the SCL line of the bus at the high level, Master 1 starts counting a clock pulse in the high level. Then, the master, which has finished the counting a clock pulse in the high level, pulls down the SCL pin to the low level.

The clock pulse on the bus is determined by the master device with the shortest high-level period and the master device with the longest low-level period from among those master devices connected to the bus.

20.4.5 Master/slave selection

To set a master device, SBI0CR2<MST> should be set to "1".

To set a slave device, SBI0CR2<MST> should be cleared to "0". When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<MST> is cleared to "0" by the hardware.

20.4.6 Transmitter/receiver selection

To set the device as a transmitter, SBI0CR2<TRX> should be set to "1". To set the device as a receiver, SBI0CR2<TRX> should be cleared to "0".

For the I^2C bus data transfer in the slave mode, SBI0CR2<TRX> is set to "1" by the hardware if the direction bit ($R/\overline{W}$) sent from the master device is "1", and is cleared to "0" if the bit is "0".

In the master mode, after an acknowledge signal is returned from the slave device, SBI0CR2<TRX> is cleared to "0" by hardware if a transmitted direction bit is "1", and is set to "1" by hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

When a stop condition on the bus or an arbitration lost is detected, SBI0CR2<TRX> is cleared to "0" by the hardware. Table 20-3 shows SBI0CR2<TRX> changing conditions in each mode and SBI0CR2<TRX> value after changing.

Note: When SBI0CR1<NOACK> is "1", the slave address match detection and the GENERAL CALL detection are disabled, and thus SBI0CR2<TRX> remains unchanged.

Table 20-3 SBI0CR1<TRX> Operation in Each Mode

Mode	Direction bit	Changing condition	TRX after changing
Slave mode	"0"	A received slave address is the same as the value set to I2C0AR<SA>	"0"
	"1"		"1"
Master mode	"0"	ACK signal is returned	"1"
	"1"		"0"

When the serial bus interface circuit operates in the free data format, a slave address and a direction bit are not recognized. They are handled as data just after generating the start condition. SBI0CR2<TRX> is not changed by the hardware.

Downloaded from Arrow.com.

20.4.7 Start/stop condition generation

When SBI0SR2<BB> is "0", a slave address and a direction bit which are set to the SBI0DBR are output on a bus after generating a start condition by writing "1" to SBI0CR2 <MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN>. It is necessary to set SBI0CR1<ACK> to "1" before generating the start condition.

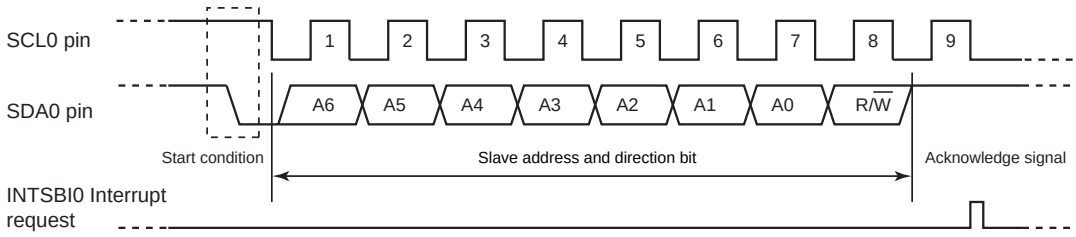


Figure 20-9 Generating the Start Condition and a Slave Address

When SBI0CR2<BB> is "1", the sequence of generating the stop condition on the bus is started by writing "1" to SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<PIN> and writing "0" to SBI0CR2<BB>.

When a stop condition is generated. The SCL line on a bus is pulled down to the low level by another device, a stop condition is generated after releasing the SCL line.

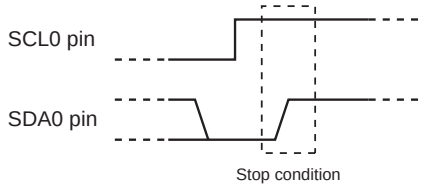


Figure 20-10 Stop Condition Generation

The bus condition can be indicated by reading the contents of SBI0SR2<BB>. SBI0SR2<BB> is set to "1" when the start condition on the bus is detected (Bus Busy State) and is cleared to "0" when the stop condition is detected (Bus Free State).

20.4.8 Interrupt service request and release

When a serial bus interface circuit is in the master mode and transferring a number of clocks set by SBI0CR1<BC> and SBI0CR1<ACK> is complete, a serial bus interface interrupt request (INTSBI0) is generated.

In the slave mode, a serial bus interface interrupt request (INTSBI0) is generated when the above and following conditions are satisfied:

- At the end of the acknowledge signal when the received slave address matches to the value set by the I2C0AR<SA> with SBI0CR1<NOACK> set at "0"
- At the end of the acknowledge signal when a "GENERAL CALL" is received with SBI0CR1<NOACK> set at "0"
- At the end of transferring or receiving after matching of the slave address or receiving of "GENERAL CALL"

When a serial bus interface interrupt request occurs, SBI0CR2<PIN> is cleared to "0". During the time that SBI0CR2<PIN> is "0", the SCL0 pin is pulled down to the low level.

Downloaded from Arrow.com.

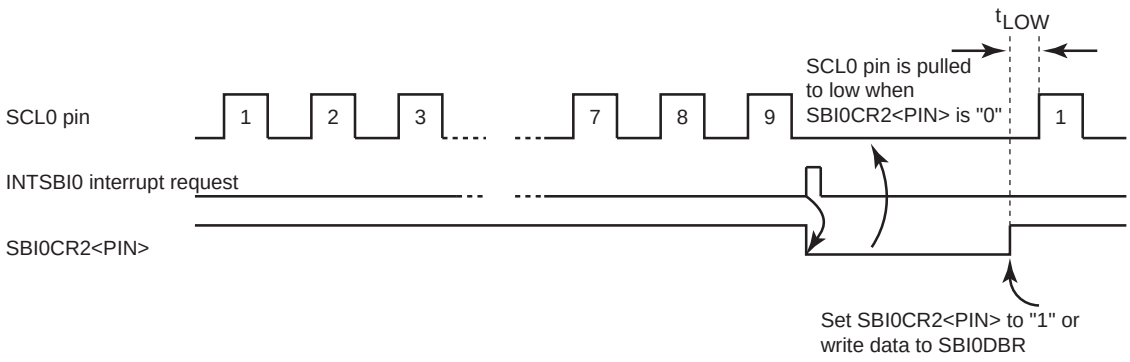


Figure 20-11 SBI0CR2<PIN> and SCL0 Pin

Writing data to SBI0DBR sets SBI0CR2<PIN> to "1". The time from SBI0CR2<PIN> being set to "1" until the SBI0 pin is released takes t_{LOW}.

Although SBI0CR2<PIN> can be set to "1" by the software, SBI0CR2<PIN> can not be cleared to "0" by the software.

20.4.9 Setting of serial bus interface mode

SBI0CR2<SBIM> is used to set serial bus interface mode.

Setting SBI0CR2<SBIM> to "1" selects the serial bus interface mode. Setting it to "0" selects the port mode.

Set SBI0CR2<SBIM> to "1" in order to set serial bus interface mode. Before setting of serial bus interface mode, confirm serial bus interface pins in a high level, and then, write "1" to SBI0CR2<SBIM>.

And switch a port mode after confirming that a bus is free and set SBI0CR2<SBIM> to "0".

Note: When SBI0CR2<SBIM> is "0", no data can be written to SBI0CR2 except SBI0CR2<SBIM>. Before setting values to SBI0CR2, write "1" to SBI0CR2<SBIM> to activate the serial bus interface mode.

20.4.10 Software reset

The serial bus interface circuit has a software reset function that initializes the serial bus interface circuit. If the serial bus interface circuit locks up, for example, due to noise, it can be initialized by using this function.

A software reset is generated by writing "10" and then "01" to SBI0CR2<SWRST>.

After a software reset is generated, the serial bus interface circuit is initialized and all the bits of SBI0CR2 register, except SBI0CR2<SBIM> and the SBI0CR1, I2C0AR<SA> and SBI0SR2 registers, are initialized.

20.4.11 Arbitration lost detection monitor

Since more than one master device can exist simultaneously on a bus, a bus arbitration procedure is implemented in order to guarantee the contents of transferred data.

Data on the SDA line is used for bus arbitration of the I^2C bus.

The following shows an example of a bus arbitration procedure when two master devices exist simultaneously on a bus. Master 1 and Master 2 output the same data until point "a". After that, when Master 1 outputs "1" and Master 2 outputs "0", since the SDA line of a bus is wired AND, the SDA line is pulled down to the low level by Master 2. When the SCL line of a bus is pulled-up at point "b", the slave device reads data on the SDA line, that is data in Master 2. Data transmitted from Master 1 becomes invalid. The state in Master

Downloaded from Arrow.com.

1 is called "arbitration lost". A master device which loses arbitration releases the SDA pin and the SCL pin in order not to effect data transmitted from other masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

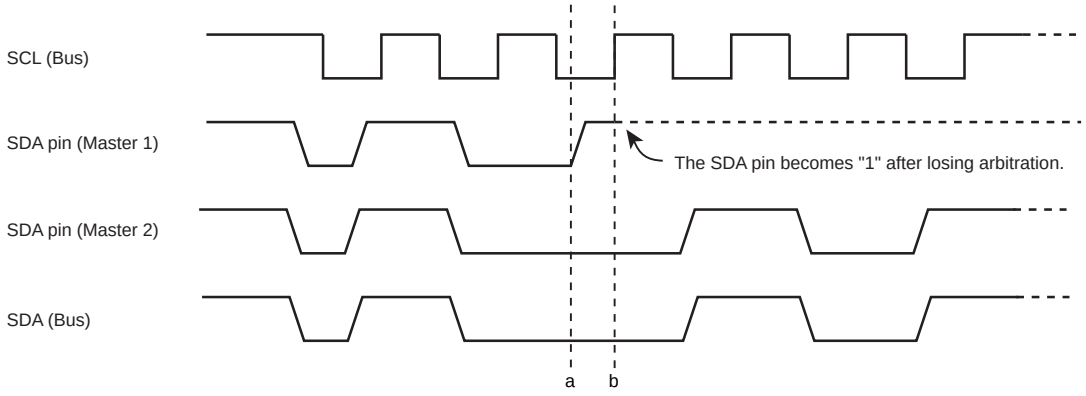


Figure 20-12 Arbitration Lost

The serial bus interface circuit compares levels of a SDA line of a bus with its SDA pin at the rising edge of the SCL line. If the levels are unmatched, arbitration is lost and SBI0SR2<AL> is set to "1".

When SBI0SR2<AL> is set to "1", SBI0CR2<MST> and SBI0CR2<TRX> are cleared to "0" and the mode is switched to a slave receiver mode. Thus, the serial bus interface circuit stops output of clock pulses during data transfer after the SBI0SR2<AL> is set to "1". After the data transfer is completed, SBICR2<PIN> is cleared to "0" and the SCL pin is pulled down to the low level.

SBI0SR2<AL> is cleared to "0" by writing data to the SBI0DBR, reading data from the SBI0DBR or writing data to the SBI0CR2.

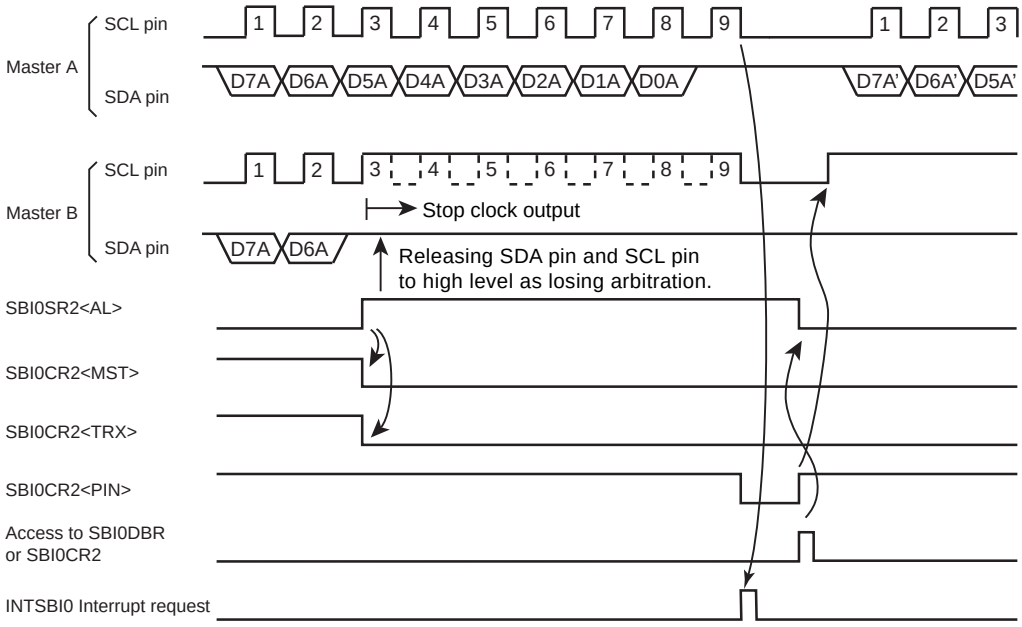


Figure 20-13 Example When Master B is a Serial Bus Interface Circuit

Downloaded from Arrow.com.

20.4.12 Slave address match detection monitor

In the slave mode, SBI0SR2<AAS> is set to "1" when the received data is "GENERAL CALL" or the received data matches the slave address setting by I2C0AR<SA> with SBI0CR1<NOACK> set at "0" and the I^2C bus mode is active (I2C0AR<ALS>="0").

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AAS> remains at "0" even if a "GENERAL CALL" is received or the same slave address as the I2C0AR<SA> set value is received.

When a serial bus interface circuit operates in the free data format (I2C0AR<ALS>= "1"), SBI0SR2<AAS> is set to "1" after receiving the first 1-word of data. SBI0SR2<AAS> is cleared to "0" by writing data to the SBI0DBR or reading data from the SBI0DBR.

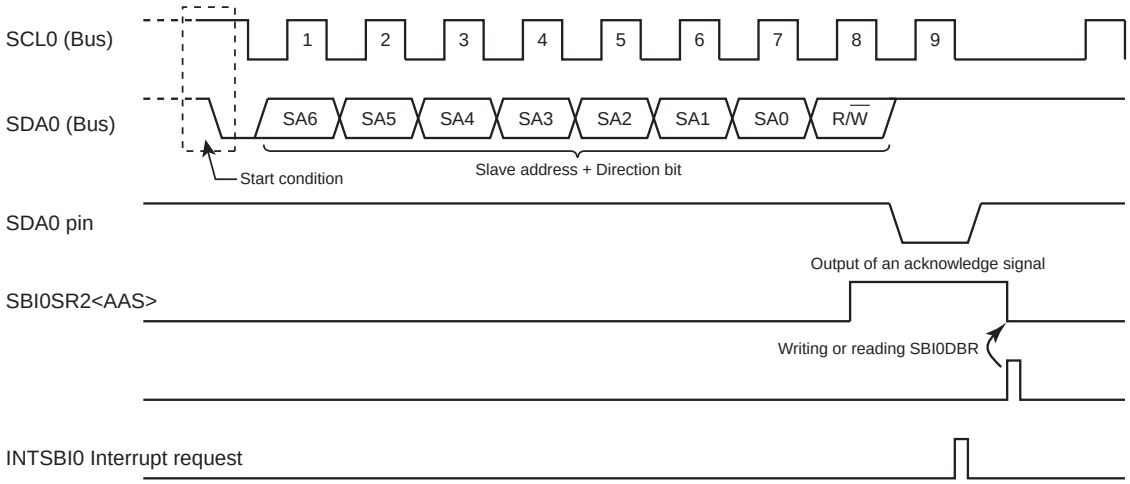


Figure 20-14 Changes in the Slave Address Match Detection Monitor

20.4.13 GENERAL CALL detection monitor

SBI0SR2<AD0> is set to "1" when SBI0CR1<NOACK> is "0" and GENERAL CALL (all 8-bit received data is "0" immediately after a start condition) in a slave mode.

Setting SBI0CR1<NOACK> to "1" disables the subsequent slave address match and GENERAL CALL detections. SBI0SR2<AD0> remains at "0" even if a "GENERAL CALL" is received.

SBI0SR2<AD0> is cleared to "0" when a start or stop condition is detected on a bus.

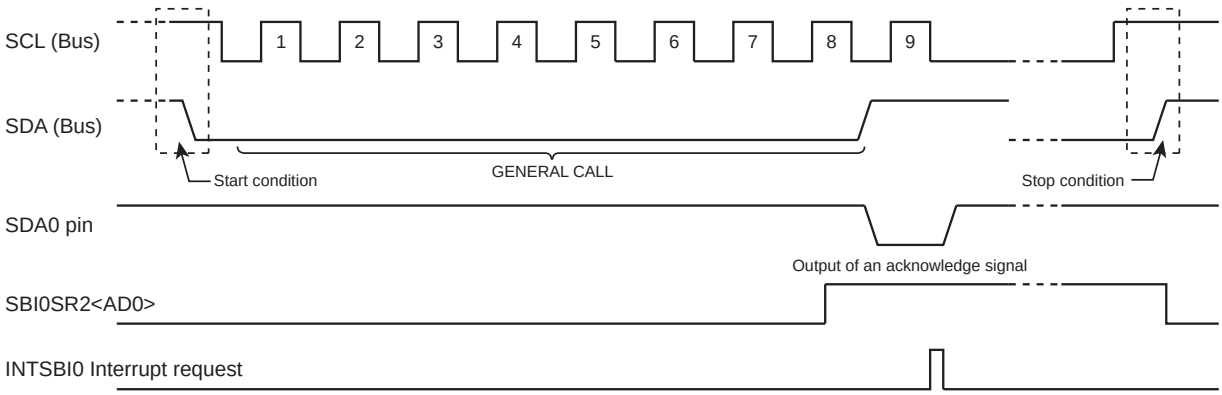


Figure 20-15 Changes in the GENERAL CALL Detection Monitor

Downloaded from Arrow.com.

20.4.14 Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to SBI0SR2<LRB>.

In the acknowledge mode, immediately after an interrupt request is generated, an acknowledge signal is read by reading the contents of SBI0SR2<LRB>.

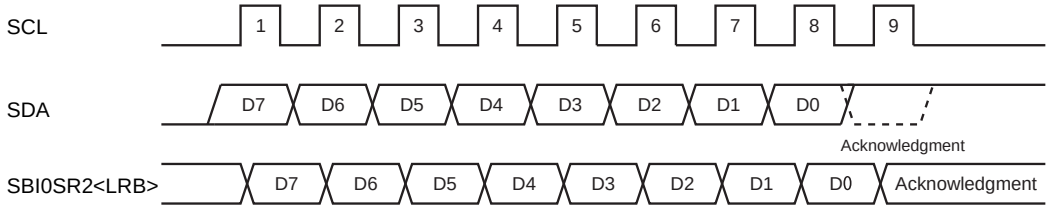


Figure 20-16 Changes in the Last Received Bit Monitor

20.4.15 Slave address and address recognition mode specification

When the serial bus interface circuit is used in the I^2C bus mode, clear I2C0AR<ALS> to "0", and set I2C0AR<SA> to the slave address.

When the serial bus interface circuit is used with a free data format not to recognize the slave address, set I2C0AR<ALS> to "1". With a free data format, the slave address and the direction bit are not recognized, and they are processed as data from immediately after the start condition.

Downloaded from Arrow.com.

20.5 Data Transfer of I²C Bus

20.5.1 Device initialization

Set POFFCR1<SBI0EN> to "1".

After confirming that the serial bus interface pin is high level, set SBI0CR2<SBIM> to "1" to select the serial bus interface mode.

Set SBI0CR1<ACK> to "1", SBI0CR1<NOACK> to "0" and SBI0CR1<BC> to "000" to count the number of clocks for an acknowledge signal, to enable the slave address match detection and the GENERAL CALL detection, and set the data length to 8 bits. Set T_{HIGH} and T_{LOW} at SBI0CR1<SCK>.

Set a slave address at I2C0AR<SA> and set I2C0AR<ALS> to "0" to select the I²C bus mode.

Finally, set SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<BB> to "0", SBI0CR2<PIN> to "1" and SBI0CR2<SWRST> to "00" for specifying the default setting to a slave receiver mode.

Note: The initialization of a serial bus interface circuit must be complete within the time from all devices which are connected to a bus have initialized to and device does not generate a start condition. If not, the data can not be received correctly because the other device starts transferring before an end of the initialization of a serial bus interface circuit.

Example :Initialize a device

```
CHK_PORT:  LD    A, (P8PRD)              ; Checks whether the serial bus interface pin is at the high level
           AND   A, 0x03
           CMP   A, 0x03
           JR    NZ, code_addr(CHK_PORT)
           LD    (SBI0CR2), 0x18         ; Selects the serial bus interface mode
           LD    (SBI0CR1), 0x16         ; Selects the acknowledgment mode and sets SBI0CR1<SCK> to "110"
           LD    (I2C0AR), 0xa0          ; Sets the slave address to 1010000 and selects the I2C bus mode
           LD    (SBI0CR2), 0x18         ; Selects the slave receiver mode
```

20.5.2 Start condition and slave address generation

Confirm a bus free status (SBI0SR2<BB>="0").

Set SBI0CR1<ACK> to "1" and specify a slave address and a direction bit to be transmitted to the SBI0DBR.

By writing "1" to SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN>, the start condition is generated on a bus and then, the slave address and the direction bit which are set to the SBI0DBR are output. The time from generating the START condition until the falling SBI0 pin takes t^{HIGH}.

An interrupt request occurs at the 9th falling edge of a SCL clock cycle, and SBI0CR2<PIN> is cleared to "0". The SCL0 pin is pulled down to the low level while SBI0CR2<PIN> is "0". When an interrupt request occurs, SBI0CR2<TRX> changes by the hardware according to the direction bit only when an acknowledge signal is returned from the slave device.

Note 1: Do not write a slave address to the SBI0DBR while data is transferred. If data is written to the SBI0DBR, data to be output may be destroyed.

Note 2: The bus free state must be confirmed by software within 98.0 μs (the shortest transmitting time according to the standard mode I²C bus standard) or 23.7μs (the shortest transmitting time according to the fast mode I²C bus standard) after setting of the slave address to be output. Only when the bus free state is confirmed, set "1" to SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> to generate the start conditions. If the writing of slave address and setting of SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> doesn't finish within 98.0μs or 23.7μs, the other masters may start the transferring and the slave address data written in SBI0DBR may be broken.

Downloaded from Arrow.com.

Example :Generate the start condition

```
CHK_BB:   TEST   (SBI0SR2).BB          ; Confirms that the bus is free
          JR     F, code_addr(CHK_BB)
          LD     (SBI0DBR), 0xcb       ; The transmission slave address 0x65 and the direction bit "1"
          LD     (SBI0CR2), 0xf8       ; Write "1" to SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

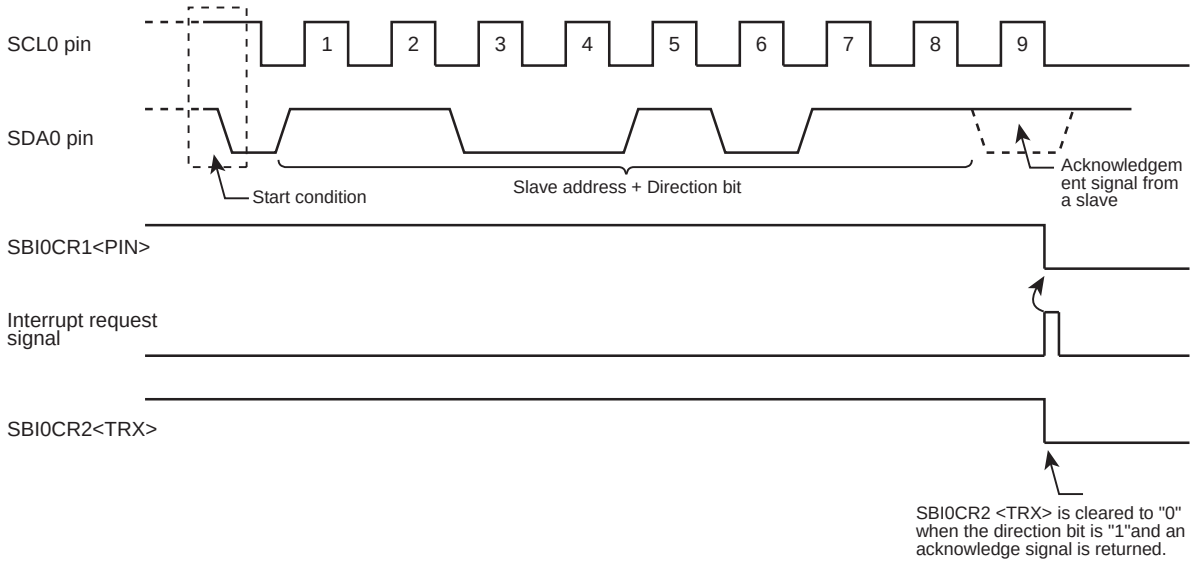


Figure 20-17 Generating the Start Condition and the Slave Address

20.5.3 1-word data transfer

Check SBI0SR2<MST> by the interrupt process after a 1-word data transfer is completed, and determine whether the mode is a master or slave.

20.5.3.1 When SBI0SR2<MST> is "1" (Master mode)

Check SBI0SR2<TRX> and determine whether the mode is a transmitter or receiver.

(1) When SBI0SR2<TRX> is "1" (Transmitter mode)

Check SBI0SR2<LRB>. When SBI0SR2<LRB> is "1", a receiver does not request data. Implement the process to generate a stop condition (described later) and terminate data transfer.

When SBI0SR2<LRB> is "0", the receiver requests subsequent data. When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC> again, set SBI0CR1<ACK> to "1", and write the transmitted data to SBI0DBR.

After writing the data, SBI0CR2<PIN> becomes "1", a serial clock pulse is generated for transferring the subsequent 1-word data from the SCL0 pin, and then the 1-word data is transmitted from the SDA0 pin.

After the data is transmitted, an interrupt request occurs. SBI0CR2<PIN> become "0" and the SCL0 pin is set to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the SBI0SR2<LRB> checking above.

Downloaded from Arrow.com.

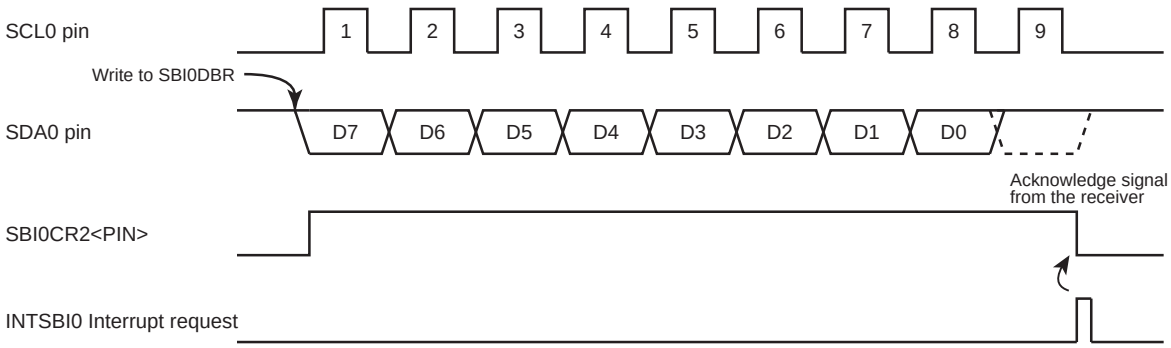


Figure 20-18 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1"

(2) When SBI0SR2<TRX> is "0" (Receiver mode)

When the data to be transmitted subsequently is other than 8 bits, set SBI0CR1<BC> again. Set SBI0CR1< ACK> to "1" and read the received data from the SBI0DBR (Reading data is undefined immediately after a slave address is sent).

After the data is read, SBI0CR2<PIN> becomes "1" by writing the dummy data (0x00) to the SBI0DBR. The serial bus interface circuit outputs a serial clock pulse to the SCL0 pin to transfer the subsequent 1-word data and sets the SDA0 pin to "0" at the acknowledge signal timing.

An interrupt request occurs and SBI0CR2<PIN> becomes "0". Then a serial bus interface circuit outputs a clock pulse for 1-word data transfer and the acknowledge signal by writing data to the SBI0DBR or setting SBI0CR2<PIN> to "1" after reading the received data.

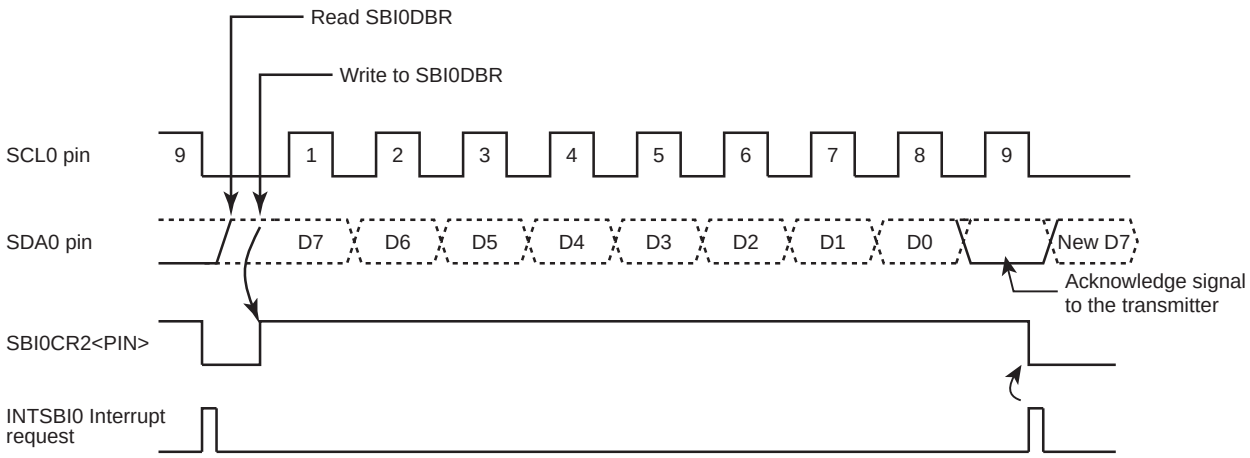


Figure 20-19 Example when SBI0CR1<BC>="000" and SBI0CR1<ACK>="1"

To make the transmitter terminate transmission, execute following procedure before receiving a last data.

1. Read the received data.
2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "000".
3. To set SBI0CR2<PIN> to "1", write a dummy data (0x00) to SBI0DBR.

Transfer 1-word data in which no clock is generated for an acknowledge signal by setting SBI0CR2<PIN> to "1". Next, execute following procedure.

Downloaded from Arrow.com.

1. Read the received data.

2. Clear SBI0CR1<ACK> to "0" and set SBI0CR1<BC> to "001".

3. To set SBI0CR2<PIN> to "1", write a dummy data (0x00) to SBI0DBR.

Transfer 1-bit data by setting SBI0CR1<PIN> to "1".

In this case, since the master device is a receiver, the SDA line on a bus keeps the high level. The transmitter receives the high-level signal as a negative acknowledge signal. The receiver indicates to the transmitter that data transfer is complete.

After 1-bit data is received and an interrupt request has occurred, generate the stop condition to terminate data transfer.

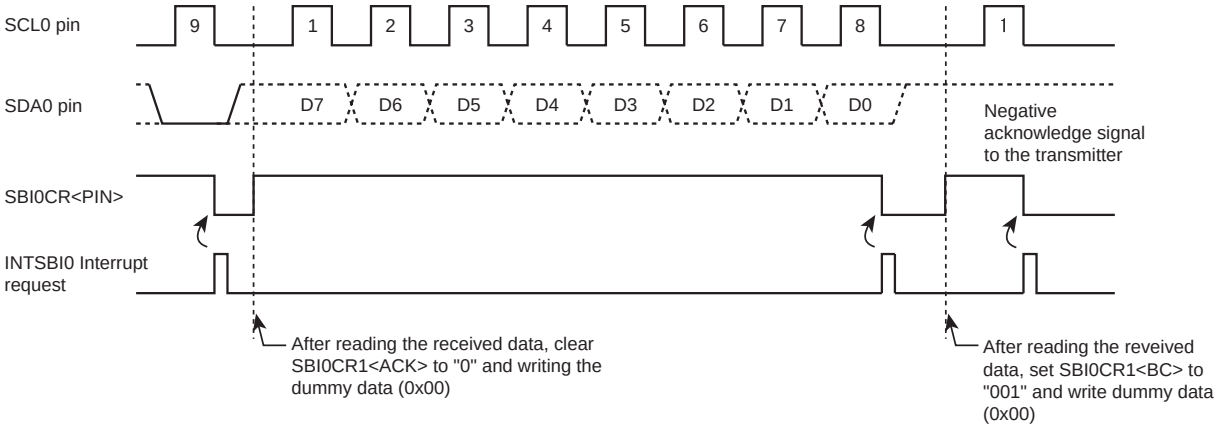


Figure 20-20 Termination of Data Transfer in the Master Receiver Mode

20.5.3.2 When SBI0SR2<MST> is "0" (Slave mode)

In the slave mode, a serial bus interface circuit operates either in the normal slave mode or in the slave mode after losing arbitration.

In the slave mode, the conditions of generating the serial bus interface interrupt request (INTSBI0) are follows:

- At the end of the acknowledge signal when the received slave address matches the value set by the I2C0AR<SA> with SBI0CR1<NOACK> set at "0"
- At the end of the acknowledge signal when a "GENERAL CALL" is received with SBI0CR1<NOACK> set at "0"
- At the end of transferring or receiving after matching of slave address or receiving of "GENERAL CALL"

The serial bus interface circuit changes to the slave mode if arbitration is lost in the master mode. And an interrupt request occurs when the word data transfer terminates after losing arbitration. The generation of the interrupt request and the behavior of SBI0CR2<PIN> after losing arbitration are shown in Table 20-4.

Downloaded from Arrow.com.

Table 20-4 The Behavior of an interrupt request and SBI0CR2<PIN> After Losing Arbitration

	When the Arbitration Lost Occurs during Transmission of Slave Address as a Master	When the Arbitration Lost Occurs during Transmission of Data as Master Transmitter
interrupt request	An interrupt request is generated at the termination of word-data transfer.	
SBI0CR2<PIN>	SBI0CR2<PIN> is cleared to "0".	

When an interrupt request occurs, SBI0CR2<PIN> is reset to "0", and the SCL0 pin is set to the low level. Either writing data to the SBI0DBR or setting SBI0CR2<PIN> to "1" releases the SCL0 pin after taking t_{LOW}.

Check SBI0SR2<AL>, SBI0SR2<TRX>, SBI0SR2<AAS> and SBI0SR2<AD0> and implement processes according to conditions listed in Table 20-5.

Table 20-5 Operation in the Slave Mode

SBI0SR2<TRX>	SBI0SR2<AL>	SBI0SR2<AAS>	SBI0SR2<AD0>	Conditions	Process
1	1	1	0	The serial bus interface circuit loses arbitration when transmitting a slave address, and receives a slave address of which the value of the direction bit sent from another master is "1".	Set the number of bits in 1 word to SBI0CR1<BC> and write the transmitted data to the SBI0DBR.
	0	1	0	In the slave receiver mode, the serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "1".	
		0	0	In the slave transmitter mode, the serial bus interface circuit finishes the transmission of 1-word data	Check SBI0SR2<LRB>. If it is set to "1", set SBI0CR2<PIN> to "1" since the receiver does not request subsequent data. Then, clear SBI0CR2<TRX> to "0" to release the bus. If SBI0SR2<LRB> is set to "0", set the number of bits in 1 word to SBI0CR1<BC> and write the transmitted data to SBI0DBR since the receiver requests subsequent data.
0	1	1	1/0	The serial bus interface circuit loses arbitration when transmitting a slave address, and receives a slave address of which the value of the direction bit sent from another master is "0" or receives a "GENERAL CALL".	Write the dummy data (0x00) to the SBI0DBR to set SBI0CR2<PIN> to "1", or write "1" to SBI0CR2<PIN>.
		0	0	The serial bus interface circuit loses arbitration when transmitting a slave address or data, and terminates transferring the word data.	The serial bus interface circuit is changed to the slave mode. Write the dummy data (0x00) to the SBI0DBR to clear SBI0SR2<AL> to "0" and set SBI0CR2<PIN> to "1".
	0	1	1/0	In the slave receiver mode, the serial bus interface circuit receives a slave address of which the value of the direction bit sent from the master is "0" or receives "GENERAL CALL".	Write the dummy data (0x00) to the SBI0DBR to set SBI0CR2<PIN> to "1", or write "1" to SBI0CR2<PIN>.
		0	1/0	In the slave receiver mode, the serial bus interface circuit terminates the receipt of 1-word data.	Set the number of bits in 1-word to SBI0CR1<BC>, read the received data from the SBI0DBR and write the dummy data (0x00).

Note: In the slave mode, if the slave address set in I2C0AR<SA> is "0x00", a START Byte "0x01" in I²C bus standard is received, the device detects slave address match and SBI0CR2<TRX> is set to "1". Do not set I2C0AR<SA> to "0x00".

Downloaded from Arrow.com.

20.5.4 Stop condition generation

When SBI0CR2<BB> is "1", a sequence of generating a stop condition is started by setting "1" to SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<PIN> and clearing SBI0CR2<BB> to "0". Do not modify the contents of SBI0CR2<MST>, SBI0CR2<TRX>, SBI0CR2<BB> and SBI0CR2<PIN> until a stop condition is generated on a bus.

When a SCL line on a bus is pulled down by other devices, a serial bus interface circuit generates a stop condition after a SCL line is released.

The time from the releasing SCL line until the generating the STOP condition takes t_{HIGH}.

Example :Generate the stop condition

```
        LD      (SBI0CR2), 0xD8      ; Sets SBI0CR2<MST>, <TRX> and <PIN> to "1" and SBI0CR2<BB> to "0"
CHK_BB: TEST    (SBI0SR2).BB         ;Waits until the bus is set free
        JR      T, code_addr(CHK_BB)
```

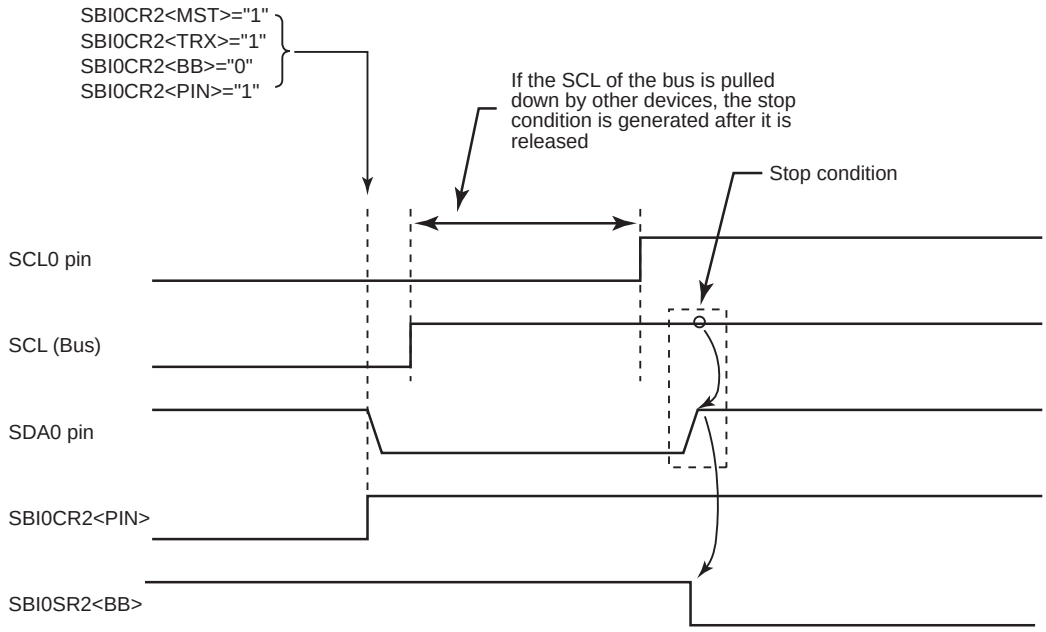


Figure 20-21 Stop Condition Generation

20.5.5 Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart the serial bus interface circuit.

Clear SBI0CR2<MST>, SBI0CR2<TRX> and SBI0CR2<BB> to "0" and set SBI0CR2 <PIN> to "1". The SDA0 pin retains the high level and the SCL0 pin is released.

Since this is not a stop condition, the bus is assumed to be in a busy state from other devices.

Check SBI0SR2<BB> until it becomes "0" to check that the SCL0 pin of the serial bus interface circuit is released.

Check SBI0SR2<LRB> until it becomes "1" to check that the SCL line on the bus is not pulled down to the low level by other devices.

Downloaded from Arrow.com.

After confirming that the bus stays in a free state, generate a start condition in the procedure "20.5.2 Start condition and slave address generation".

In order to meet the setup time at a restart, take at least 4.7μs of waiting time by the software in the standard mode I²C bus standard or at least 0.6μs of waiting time in the fast mode I²C bus standard from the time of restarting to confirm that a bus is free until the time to generate a start condition.

Note: When the master is in the receiver mode, it is necessary to stop the data transmission from the slave device before the STOP condition is generated. To stop the transmission, the master device make the slave device receiving a negative acknowledge. Therefore, SBI0SR2<LRB> is "1" before generating the Restart and it can not be confirmed that SCL line is not pulled down by other devices. Please confirm the SCL line state by reading the port.

Example :Generate a restart

```
            LD      (SBI0CR2), 0x18        ; Sets SBI0CR2<MST>, <TRX> and <BB> to "0" and SBI0CR2<PIN> to "1"
CHK_BB:     TEST    (SBI0SR2).BB           ; Waits until SBI0SR2<BB> becomes "0"
            JR      T, code_addr(CHK_BB)
CHK_LRB:    TEST    (SBI0SR2).LRB          ; Waits until SBI0SR2<LRB> becomes "1"
            JR      F, code_addr(CHK_LRB)
                    .
                    .                      ; Wait time process by the software
                    .
            LD      (SBI0CR2), 0xf8        ; Sets SBI0CR2<MST>, <TRX>, <BB> and <PIN> to "1"
```

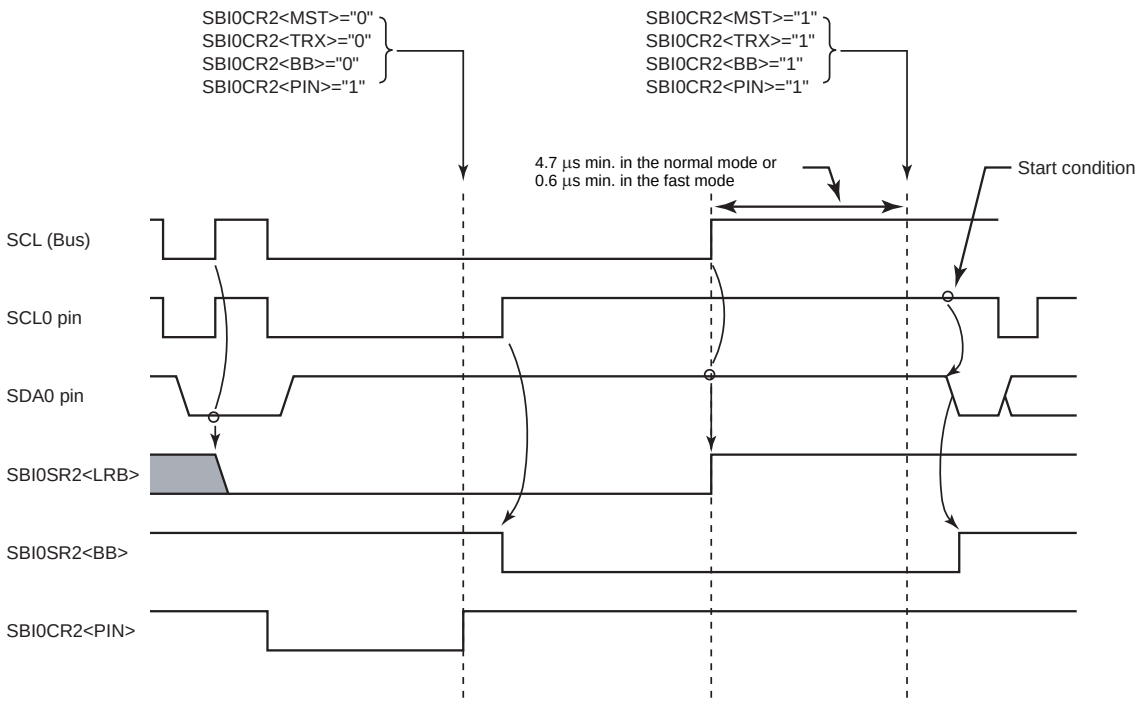


Figure 20-22 Timing Diagram When Restarting

Downloaded from Arrow.com.

20.6 AC Specifications

The AC specifications are as listed below.

The operating mode (fast or standard) mode should be selected suitable for frequency of fcgck. For these operating mode, refer to the following table.

Table 20-6 AC Specifications (Circuit Output Timing)

Parameter	Symbol	Standard mode		Fast mode		Unit
		MIN.	MAX.	MIN.	MAX.	
SCL clock frequency	f_{SCL}	0	fcgck / (m+n)	0	fcgck / (m+n)	kHz
Hold time (re)start condition. This period is followed by generation of the first clock pulse.	$t_{HD;STA}$	m / fcgck	-	m / fcgck	-	μs
Low-level period of SCL clock (output)	t_{LOW}	n / fcgck	-	n / fcgck	-	μs
High-level period of SCL clock (output)	t_{HIGH}	m / fcgck	-	m / fcgck	-	μs
Low-level period of SCL clock (input)	t_{LOW}	5 / fcgck	-	5 / fcgck	-	μs
High-level period of SCL clock (input)	t_{HIGH}	3 / fcgck	-	3 / fcgck	-	μs
Restart condition setup time	$t_{SU;STA}$	Depends on the software	-	Depends on the software	-	μs
Data hold time	$t_{HD;DAT}$	0	5 / fcgck	0	5 / fcgck	μs
Data setup time	$t_{SU;DAT}$	250	-	100	-	ns
Rising time of SDA and SCL signals	t_r	-	1000	-	300	ns
Falling time of SDA and SCL signals	t_f	-	300	-	300	ns
Stop condition setup time	$t_{SU;STO}$	m / fcgck	-	m / fcgck	-	μs
Bus free time between the stop condition and the start condition	t_{BUF}	Depends on the software	-	Depends on the software	-	μs
Time before rising of SCL after SBICR2<PIN> is changed from "0" to "1"	$t_{SU;SCL}$	n / fcgck	-	n / fcgck	-	μs

Note: For m and n, refer to"20.4.4.1 Clock source".

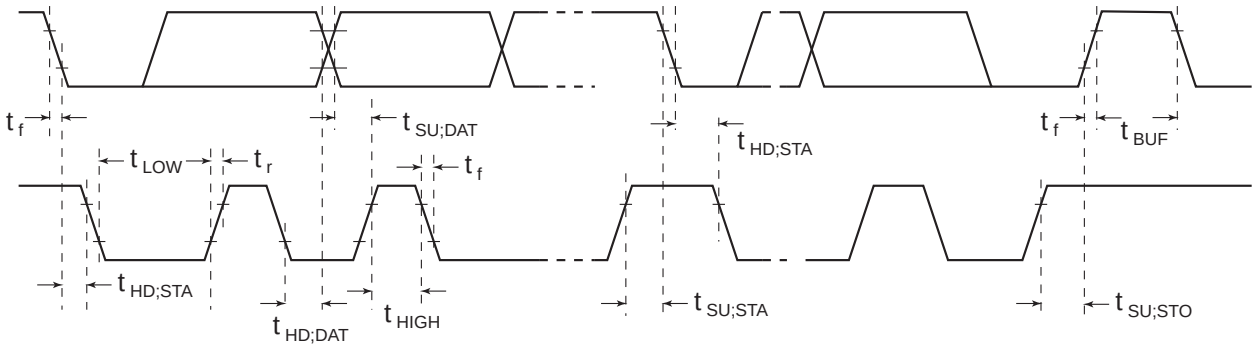


Figure 20-23 Definition of Timing (No. 1)

Downloaded from Arrow.com.

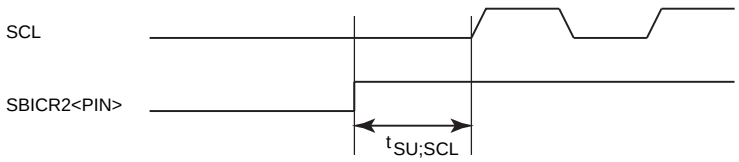


Figure 20-24 Definition of Timing (No. 2)

Downloaded from Arrow.com.

21. Key-on Wakeup (KWU)

The key-on wakeup is a function for releasing the STOP mode at the $\overline{\text{STOP}}$ pin or at pins KWI2 through KWI0.

21.1 Configuration

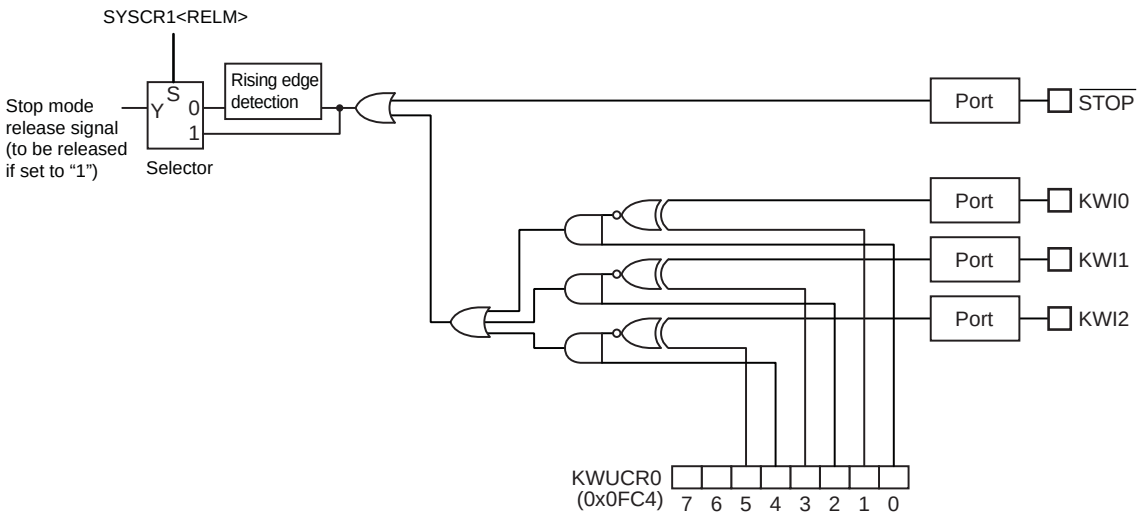


Figure 21-1 Key-on Wakeup Circuit

Downloaded from Arrow.com.

21.2 Control

Key-on wakeup control registers (KWUCR0) can be configured to designate the key-on wakeup pins (KWI2 through KWI0) as STOP mode release pins and to specify the STOP mode release levels of each of these designated pins.

Key-on wakeup control register 0

KWUCR0		7	6	5	4	3	2	1	0
(0x00FC4)	Bit Symbol	"0"	"0"	KW2LE	KW2EN	KW1LE	KW1EN	KW0LE	KW0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

KW2LE	STOP mode release level of KWI2 pin	0: 1:	Low level High level
KW2EN	Input enable/disable control of KWI2 pin	0: 1:	Disable Enable
KW1LE	STOP mode release level of KWI1	0: 1:	Low level High level
KW1EN	Input enable/disable control of KWI1 pin	0: 1:	Disable Enable
KW0LE	STOP mode release level of KWI0 pin	0: 1:	Low level High level
KW0EN	Input enable/disable control of KWI0 pin	0: 1:	Disable Enable

Note: Make sure that you write "0" to bit7 and bit6 of KWUCR0.

Key-on wakeup control register 1

KWUCR1		7	6	5	4	3	2	1	0
(0x00FC5)	Bit Symbol	KW7LE	KW7EN	KW6LE	KW6EN	KW5LE	KW5EN	KW4LE	KW4EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

KW7LE	STOP mode release level of KWI7 pin	0: 1:	Low level High level
KW7EN	Input enable/disable control of KWI7 pin	0: 1:	Disable Enable
KW6LE	STOP mode release level of KWI6 pin	0: 1:	Low level High level
KW6EN	Input enable/disable control of KWI6 pin	0: 1:	Disable Enable
KW5LE	STOP mode release level of KWI5 pin	0: 1:	Low level High level
KW5EN	Input enable/disable control of KWI5 pin	0: 1:	Disable Enable
KW4LE	STOP mode release level of KWI4 pin	0: 1:	Low level High level
KW4EN	Input enable/disable control of KWI4 pin	0: 1:	Disable Enable

Downloaded from Arrow.com.

21.3 Functions

By using the key-on wakeup function, the STOP mode can be released at a $\overline{STOP}$ pin or at KWIm pin (m: 0 through 2). After resetting, the $\overline{STOP}$ pin is the only STOP mode release pin. To designate the KWIm pin as a STOP mode release pin, therefore, it is necessary to configure the key-on wakeup control register (KWUCRn) (n: 0 or 1). Because the $\overline{STOP}$ pin lacks a function for disabling inputs, it can be designated as a pin for receiving a STOP mode release signal, irrespective of whether the key-on wakeup function is used or not.

- Setting KWUCRn and P4PU registers

  To designate a key-on wakeup pin (KWIm) as a STOP mode release pin, set KWUCRn<KWmEN> to "1". After KWIm pin is set to "1" at KWUCRn<KWmEN>, a specific STOP mode release level can be specified for this pin at KWUCRn<KWmLE>. If KWUCRn<KWmLE> is set to "0", STOP mode is released when an input is at a low level. If it is set to "1", STOP mode is released when an input is at a high level. For example, if you want to release STOP mode by inputting a high-level signal into a KWI0 pin, set KWUCR0<KW0EN> to "1", " and KWUCR0<KW0LE> to "1".

  Each KWIm pin can be connected to internal pull-up resistors. Before connecting to internal pull-up resistors, the corresponding bits in the pull-up control register (P4PU) at port P4 must be set to "1".

- Starting STOP mode

  To start the STOP mode, set SYSCR1<RELM> to "1" (level release mode), and SYSCR1<STOP> to "1".

  To use the key-on wakeup function, do not set SYSCR1<RELM> to "0" (edge release mode). If the key-on wakeup function is used in edge release mode, STOP mode cannot be released, although a rising edge is input into the $\overline{STOP}$ pin. This is because the KWIm pin enabling inputs to be received is at a release level after the STOP mode starts.

- Releasing STOP mode

  To release STOP mode, input a high-level signal into the $\overline{STOP}$ pin or input a specific release level into the KWIm pin for which receipt of inputs is enabled. If you want to release STOP mode at the KWIm pin, rather than the $\overline{STOP}$ pin, continue inputting a low-level signal into the $\overline{STOP}$ pin throughout the period from when the STOP mode is started to when it is released.

  If the $\overline{STOP}$ pin or KWIm pin is already at a release level when the STOP mode starts, the following instruction will be executed without starting the STOP mode (with no warm-up performed).

Note 1: If an analog voltage is applied to KWIm pin for which receipt of inputs is enabled by the key-on wakeup control register (KWUCRn) setting, a penetration current will flow. Therefore, in this case, the analog voltage should be not applied to this pin.

Table 21-1 STOP Mode Release Level (edge)

Pin name	Release level (edge)		
	SYSCR1<RELM>="1" (level release mode)		SYSCR1<RELM>="0" (edge release mode)
	KWUCRn<KWmLE>="0"	KWUCRn<KWmLE>="1"	
$\overline{STOP}$	"H" level		Rising edge
KWIm	"L" level	"H" level	Don't use

Downloaded from Arrow.com.

Example :A case in which STOP mode is started with the release level of the $\overline{\text{STOP}}$ pin set to a high level and the release level of KWI0 set to a low level (connected to an internal pull-up resistor of the KWI0 pin)

```
DI                                  ; IMF←0
SET     (P4PU).4                    ; KWI0 (P44) connected to a pull-up resistor
LD      (KWUCR0), 0y00000001        ; the KWI0 pin is set to enable inputs, and its release level is set
                                    ; to a low level.
LD      (SYSCR1), 0y10100000        ; Starting in level release mode
```


Downloaded from Arrow.com.

22. 10-bit AD Converter (ADC)

The TMP89FW24A has a 10-bit successive approximation type AD converter.

22.1 Configuration

The circuit configuration of the 10-bit AD converter is shown in Figure 22-1.

It consists of control registers ADCCR1 and ADCCR2, converted value registers ADCDRL and ADCDRH, a DA converter, a sample-hold circuit, a comparator, a successive comparison circuit, etc.

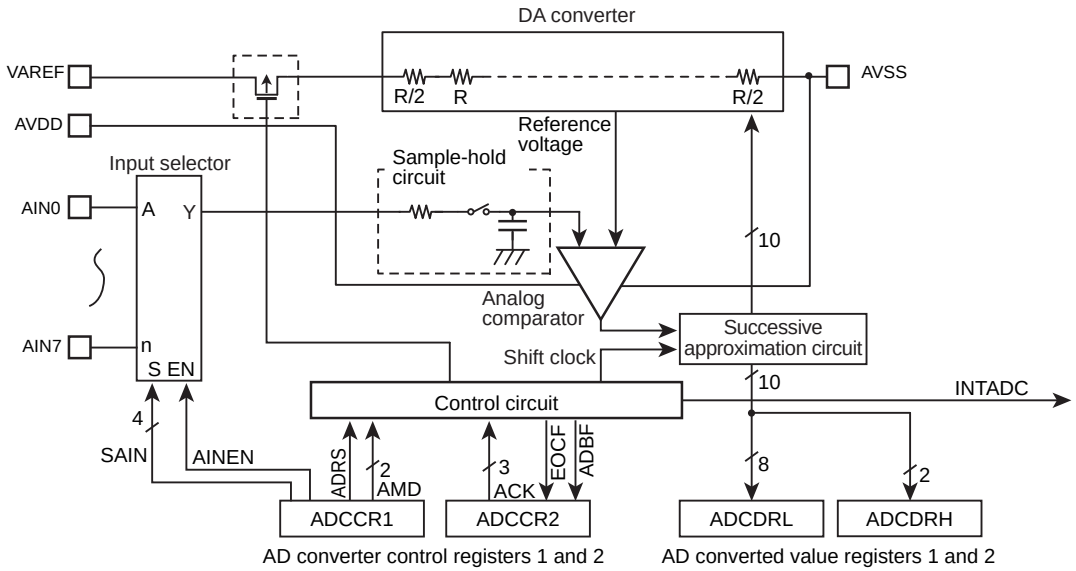


Figure 22-1 10-bit AD Converter

Note 1: Before using the AD converter, set an appropriate value to the I/O port register which is also used as an analog input port. For details, see the section on "I/O ports".

Note 2: The DA converter current (IREF) is automatically cut off at times other than during AD conversion.

Downloaded from Arrow.com.

22.2 Control

The AD converter consists of the following four registers:

1. AD converter control register 1 (ADCCR1)

   This register selects an analog channel in which to perform AD conversion, selects an AD conversion operation mode, and controls the start of the AD converter.

2. AD converter control register 2 (ADCCR2)

   This register selects the AD conversion time, and monitors the operating status of the AD converter.

3. AD converted value registers (ADCDRH and ADCDRL)

   These registers store the digital values generated by the AD converter.

2026-04-21
Downloaded from Arrow.com.

AD converter control register 1

ADCCR1 (0x00034)	7	6	5	4	3	2	1	0
Bit Symbol	ADRS	AMD		AINEN	SAIN			
Read/Write	R/W	R/W		R/W	R/W			
After reset	0	0	0	0	0	0	0	0

ADRS	AD conversion start	0: 1:	- AD conversion start
AMD	AD operating mode	00: 01: 10: 11:	AD operation disable, forcibly stop AD operation Single mode Reserved Repeat mode
AINEN	Analog input control	0: 1:	Analog input disable Analog input enable
SAIN	Analog input channel select	0000: 0001: 0010: 0011: 0100: 0101: 0110: 0111: 1000: 1001: 1010: 1011: 1100: 1101: 1110: 1111:	AIN0 AIN1 AIN2 AIN3 AIN4 AIN5 AIN6 AIN7 Reserved Reserved Reserved Reserved Reserved Reserved Reserved Reserved

Note 1: Do not perform the following operations on the ADCCR1 register while AD conversion is being executed (ADCCR2<ADBF>="1").
- Changing SAIN
- Setting AINEN to "0"
- Changing AMD (except a forced stop by setting AMD to "00")
- Setting ADRS to "1"

Note 2: If you want to disable all analog input channels, set AINEN to "0".

Note 3: Although analog input pins are also used as input/output ports, it is recommended for the purpose of maintaining the accuracy of AD conversion that you do not execute input/output instructions during AD conversion. Additionally, do not input widely varying signals into the ports adjacent to analog input pins.

Note 4: When STOP, IDLE0 or SLOW mode is started, ADRS, AMD and AINEN are initialized to "0". If you use the AD converter after returning to NORMAL mode, you must reconfigure ADRS, AMD and AINEN.

Note 5: After ADRS is set to "1", it is automatically cleared to "0". The time between when ADRS is set to "1" and when it is cleared to "0" is a maximum of 4/fcgck [s] when ADCCR2<ACK>="00*", a maximum of 15/fcgck [s] when ADCCR2<ACK>="01*", and a maximum of 52/fcgck [s] when ADCCR2<ACK>="10*".

Downloaded from Arrow.com.

AD converter control register 2

ADCCR2		7	6	5	4	3	2	1	0
(0x00035)	Bit Symbol	EOCF	ADBF	-	-	"0"	ACK		
	Read/Write	R	R	R	R	W	R/W		
	After reset	0	0	0	0	0	0	0	0

EOCF	AD conversion end flag	0: 1:	Before conversion or during conversion Conversion end
ADBF	AD conversion BUSY flag	0: 1:	AD conversion being halted AD conversion being executed
ACK	AD conversion time select (examples of AD conversion time are shown in the table below)	000: 001: 010: 011: 100: 101: 110: 111:	39/fcgck 78/fcgck 156/fcgck 312/fcgck 624/fcgck 1248/fcgck Reserved Reserved

Note 1: Make sure that you make the ACK setting when AD conversion is in a halt condition (ADCCR2<ADBF>="0").

Note 2: Make sure that you write "0" to bit 3 of ADCCR2.

Note 3: If STOP, IDLE0 or SLOW mode is started, EOCF and ADBF are initialized to "0".

Note 4: If the AD converted value register (ADCDRH) is read, EOCF is cleared to "0". It is also cleared to "0" if AD conversion is started (ADCCR1<ADRS>="1") without reading ADCDRH after completing AD conversion in single mode.

Note 5: If an instruction to read ADCCR2 is executed, 0 is read from bits 3 through 5.

Table 22-1 ACK Settings and Conversion Times Relative to Frequencies

		Frequency (fcgck)								
ACK setting	Conversion time	10MHz	8MHz	5MHz	4MHz	2.5MHz	2MHz	1MHz	0.5MHz	0.25 MHz
000	39/fcgck	-	-	-	-	15.6 μs	19.5 μs	39.0 μs	78.0 μs	156.0 μs
001	78/fcgck	-	-	15.6 μs	19.5 μs	31.2 μs	39.0 μs	78.0 μs	156.0 μs	-
010	156/fcgck	15.6 μs	19.5 μs	31.2 μs	39.0 μs	62.4 μs	78.0 μs	156.0 μs	-	-
011	312/fcgck	31.2 μs	39.0 μs	62.4 μs	78.0 μs	124.8 μs	156.0 μs	-	-	-
100	624/fcgck	62.4 μs	78.0 μs	124.8 μs	156.0 μs	-	-	-	-	-
101	1248/fcgck	124.8 μs	156.0 μs	-	-	-	-	-	-	-
11*	Reserved									

Note 1: Spaces indicated by "-" in the above table mean that it is prohibited to establish conversion times in these spaces.

fcgck: High Frequency oscillation clock [Hz]

Note 2: Above conversion times do not include the time shown below.

- Time from when ADCCR1<ADRS> is set to 1 to when AD conversion is started
- Time from when AD conversion is finished to when a converted value is stored in ADCDRL and ADCDRH.

If ACK = 00*, the longest conversion time is 10/fcgck (s). If ACK = 01*, it is 32/fcgck (s). If ACK = 10*, it is 128/fcgck (s).

Note 3: The conversion time must be longer than the following time by analog reference voltage (VAREF).

- VAREF = 4.5 to 5.5 V 15.6 μs or longer
- VAREF = 2.7 to 5.5 V 31.2 μs or longer
- VAREF = 2.2 to 5.5 V 124.8 μs or longer

Downloaded from Arrow.com.

AD converted value register (lower side)

ADCDRL		7	6	5	4	3	2	1	0
(0x00036)	Bit Symbol	AD07	AD06	AD05	AD04	AD03	AD02	AD01	AD00
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

AD converted value register (upper side)

ADCDRH		7	6	5	4	3	2	1	0
(0x00037)	Bit Symbol	-	-	-	-	-	-	AD09	AD08
	Read/Write	R	R	R	R	R	R	R	R
	After reset	0	0	0	0	0	0	0	0

Note 1: A read of ADCDRL or ADCDRH must be read after the INTADC interrupt is generated or after ADCCR2<EOCF> becomes "1".

Note 2: In single mode, do not read ADCDRL or ADCDRH during AD conversion (ADCCR2<ADBF>="1"). (If AD conversion is finished in the interim between a read of ADCDRL and a read of ADCDRH, the INTADC interrupt request is canceled, and the conversion result is lost.)

Note 3: If STOP, IDLE0 or SLOW mode is started, ADCDRL and ADCDRH are initialized to "0".

Note 4: If ADCCR1<AMD> is set to "00", ADCDRL and ADCDRH are initialized to "0".

Note 5: If an instruction to read ADCDRH is executed, "0" is read from bits 7 through 2.

Note 6: If AD conversion is finished in repeat mode in the interim between a read of ADCDRL and a read of ADCDRH, the previous converted value is retained without overwriting the AD converted value register. In this case, the INTADC interrupt request is canceled, and the conversion result is lost.

Downloaded from Arrow.com.

22.3 Functions

The 10-bit AD converter operates in either single mode in which AD conversion is performed only once or repeat mode in which AD conversion is performed repeatedly.

22.3.1 Single mode

In single mode, the voltage at a designated analog input pin is AD converted only once.

Setting ADCCR1<ADRS> to "1" after setting ADCCR1<AMD> to "01" allows AD conversion to start. ADCCR1<ADRS> is automatically cleared after the start of AD conversion. As AD conversion starts, ADCCR2<ADBF> is set to "1". It is cleared to "0" if AD conversion is finished or if AD conversion is forced to stop.

After AD conversion is finished, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2<EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. The AD converted value registers (ADCDRL and ADCDRH) should be usually read according to the INTADC interrupt processing routine. If the upper side (ADCDRH) of the AD converted value register is read, ADCCR2<EOCF> is cleared to "0".

Note: Do not perform the following operations on the ADCCR1 register when AD conversion is being executed (ADCCR2<ADBF>="1"). If the following operations are performed, there is the possibility that AD conversion may not be executed properly.

- Changing the ADCCR1<SAIN> setting
- Setting ADCCR1<AINEN> to "0"
- Changing the ADCCR1<AMD> setting (except a forced stop by setting AMD to "00")
- Setting ADCCR1<ADRS> to "1"

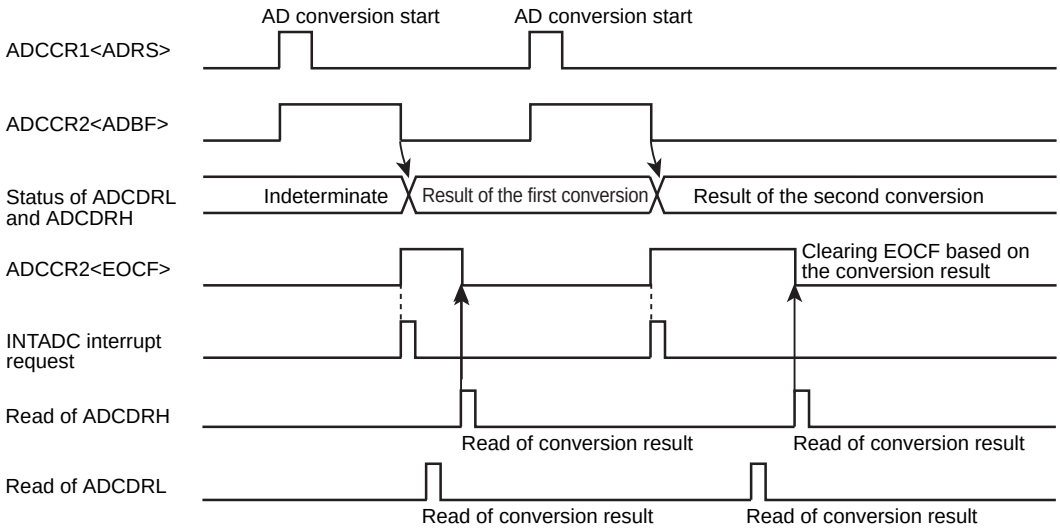


Figure 22-2 Single Mode

22.3.2 Repeat mode

In repeat mode, the voltage at an analog input pin designated at ADCCR1<SAIN> is AD converted repeatedly.

Setting ADCCR1<ADRS> to "1" after setting ADCCR1<AMD> to "11" allows AD conversion to start. After the start of AD conversion, ADCCR1<ADRS> is automatically cleared. After the first AD conversion is finished, the conversion result is stored in the AD converted value registers (ADCDRL and ADCDRH), ADCCR2<EOCF> is set to "1", and the AD conversion finished interrupt (INTADC) is generated. After this interrupt is generated, the second (next) AD conversion starts immediately.

Downloaded from Arrow.com.

The AD converted value registers (ADCDRL and ADDRH) should be read before the next AD conversion is finished. If the next AD conversion is finished in the interim between a read of ADCDRL and a read of ADCDRH, the previous converted value is retained without overwriting the AD converted value registers (ADCDRL and ADCDRH). In this case, the INTADC interrupt request is not generated, and the conversion result is lost. (See Figure 22-3.)

To stop AD conversion, write "00" (AD operation disable) to ADCCR1<AMD>. As "00" is written to ADCCR1<AMD>, AD conversion stops immediately. In this case, the converted value is not stored in the AD converted value register. As AD conversion starts, ADCCR2<ADBF> is set to "1". It is cleared to "0" if "00" is written to AMD.

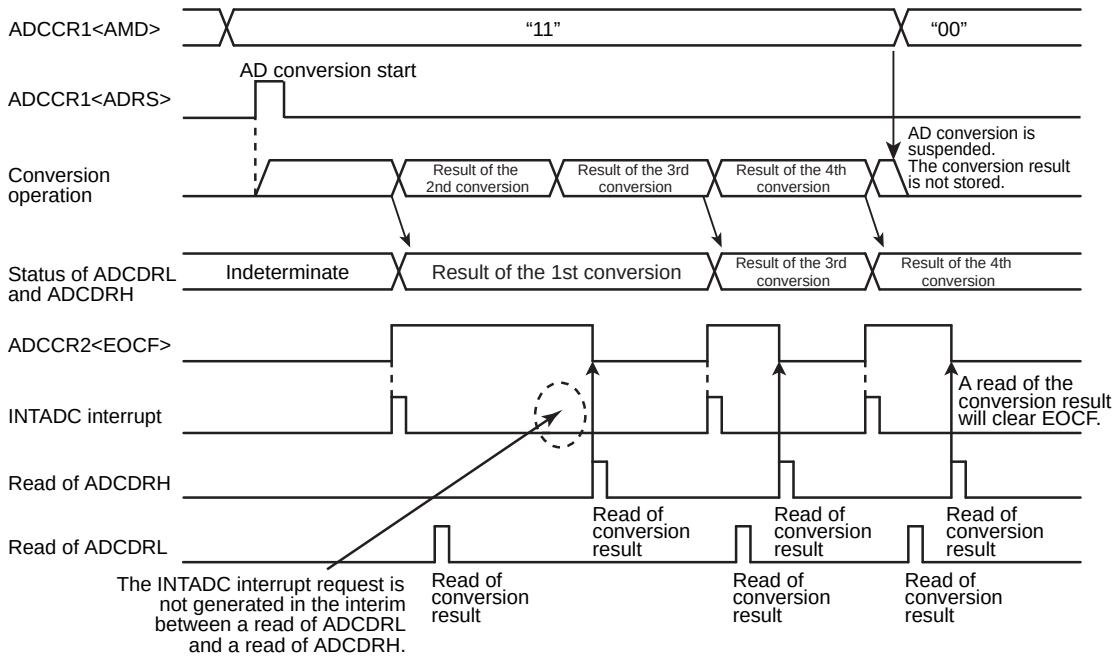


Figure 22-3 Repeat Mode

22.3.3 AD operation disable and forced stop of AD operation

If you want to force the AD converter to stop when AD conversion is ongoing in single mode or if you want to stop the AD converter when AD conversion is ongoing in repeat mode, set ADCCR1<AMD> to "00".

If ADCCR1<AMD> is set to "00", registers ADCCR2<EOCF>, ADCCR2<ADBF>, ADCDRL, and ADCDRH are initialized to "0".

Downloaded from Arrow.com.

22.4 Register Setting

1. Set the AD converter control register 1 (ADCCR1) as described below:
   - From the AD input channel select (SAIN), select the channel in which AD conversion is to be performed.
   - Set the analog input control (AINEN) to "Analog input enable".
   - At AMD, specify the AD operating mode (single or repeat mode).

2. Set the AD converter control register 2 (ADCCR2) as described below:
   - At the AD conversion time (ACK), specify the AD conversion time. For information on how to specify the conversion time, refer to the AD converter control register 2 and Table 22-1.

3. After the above two steps are completed, set "1" on the AD conversion start (ADRS) of the AD converter control register 1 (ADCCR1), and AD conversion starts immediately if single mode is selected.

4. As AD conversion is finished, the AD conversion end flag (EOCF) of the AD converter control register 2 (ADCCR2) is set to "1", the AD conversion result is stored in the AD converted value registers (ADCDRH and ADCDRL), and the INTADC interrupt request is generated.

5. After the conversion result is read from the AD converted value register (ADCDRH), EOCF is cleared to "0". EOCF will also be cleared to "0" if AD conversion is performed once again before reading the AD converted value register (ADCDRH). In this case, the previous conversion result is retained until AD conversion is finished.

Example: After selecting the conversion time 15.6 μs at 10 MHz and the analog input channel AIN3 pin, perform AD conversion once. After checking EOCF, store the conversion result in the HL register. The operation mode is single mode.

```
        : (Port setting)                        ;Before setting AD converter registers, make an appropriate port
                                                ;register setting.(For further details, refer to the section that describes
                                                ;I/O ports.)
        LD      (ADCCR1), 0y00110011            ;Select AIN3 and operation mode
        LD      (ADCCR2), 0y00000010            ;Select conversion time (156/fcgck)
        SET     (ADCCR1). 7                     ;ADRS = 1 (AD conversion start)
SLOOP :
        TEST    (ADCCR2). 7                     ;EOCF = 1 ?
        J       T, code_addr(SLOOP)
        LD      HL, (ADCDRL)                    ;Read result data
```

22.5 Starting STOP/IDLE0/SLOW Modes

If STOP/IDLE0/SLOW mode is started, registers ADCCR1<ADRS, AMD, AINEN>, ADCCR2<EOCF, ADBF>, ADCDRL and ADCDRH are initialized to "0". If any of these modes is started during AD conversion, AD conversion is suspended, and the AD converter stops (registers are likewise initialized). When restored from STOP/ IDLE0/SLOW mode, AD conversion is not automatically restarted. Therefore, registers must be reconfigured as necessary.

If STOP/IDLE0/SLOW mode is started during AD conversion, analog reference voltage is automatically disconnected and, therefore, there is no possibility of current flowing into the analog reference voltage.

Downloaded from Arrow.com.

22.6 Analog Input Voltage and AD Conversion Result

Analog input voltages correspond to AD-converted, 10-bit digital values, as shown in Figure 22-4.

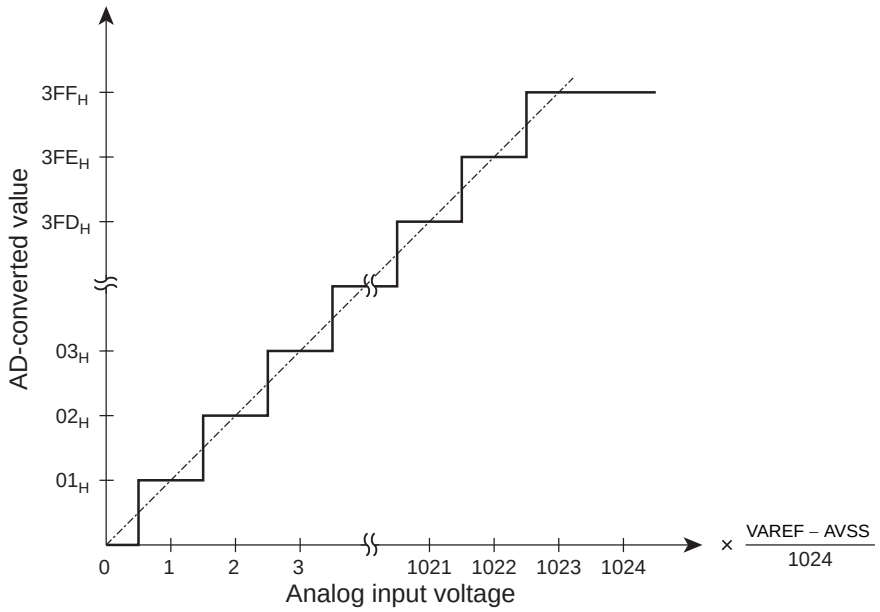


Figure 22-4 Relationships between Analog Input Voltages and AD-converted Values (typical values)

Downloaded from Arrow.com.

22.7 Precautions about the AD Converter

22.7.1 Analog input pin voltage range

Analog input pins (AIN0 through AIN7) should be used at voltages from VAREF to VSS. If any voltage outside this range is applied to one of the analog input pins, the converted value on that pin becomes uncertain, and converted values on other pins will also be affected.

22.7.2 Analog input pins used as input/output ports

Analog input pins (AIN0 to AIN7) are also used as input/output ports. In using one of analog input pins (ports) to execute AD conversion, input/output instructions at all other pins (ports) must not be executed. If they are executed, there is the possibility that the accuracy of AD conversion may deteriorate. This also applies to pins other than analog input pins; if one pin receives inputs or generates outputs, noise may occur and its adjacent pins may be affected by that noise.

22.7.3 Noise countermeasure

The internal equivalent circuit of the analog input pins is shown in Figure 22-5. The higher the output impedance of the analog input source, the more susceptible it becomes to noise. Therefore, make sure the output impedance of the signal source in your design is 5 kΩ or less. It is recommended that a capacitor be attached externally.

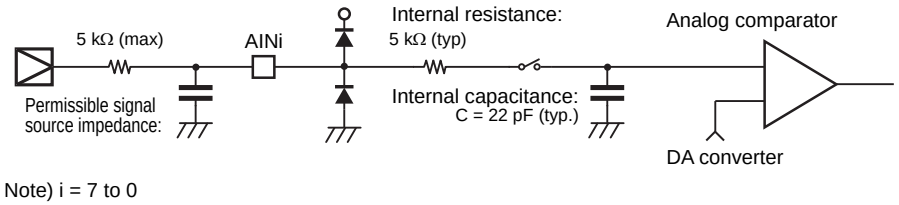


Figure 22-5 Analog Input Equivalent Circuit and Example of Input Pin Processing

Downloaded from Arrow.com.

23. LCD Driver

The TMP89FW24A has a driver and control circuit to directly drive a liquid crystal display (LCD) device. The pins to be connected to the LCD are as follows:

1. Segment output pins : 40 pins (SEG39 to SEG0)
2. Common output pins : 4 pins (COM3 to COM0)

In addition, the VLC pin is provided as a drive power supply pin, and the LV1 and LV2 pins are provided as external bleeder resistance connection pins.

Note 1: The external bleeder resistance connection pins LV1 and LV2 are shared with the segment output pins SEG38,39. When external bleeder resistance is used, SEG38,39 cannot be used for segment output.

Note 2: When the static, 1/3 or 1/2 duties are selected, unused common output pins should be opened. (It outputs bias voltage)

The LCD driver can directly drive the following five types of LCD:

1. 1/4 duty (1/3 bias) LCD Max. 160 pixels (8 segments × 20 digits)
2. 1/3 duty (1/3 bias) LCD Max. 120 pixels (8 segments × 15 digits)
3. 1/3 duty (1/2 bias) LCD Max. 120 pixels (8 segments × 15 digits)
4. 1/2 duty (1/2 bias) LCD Max. 80 pixels (8 segments × 10 digits)
5. Static LCD Max. 40 pixels (8 segments × 5 digits)

Downloaded from Arrow.com.

23.1 Configuration

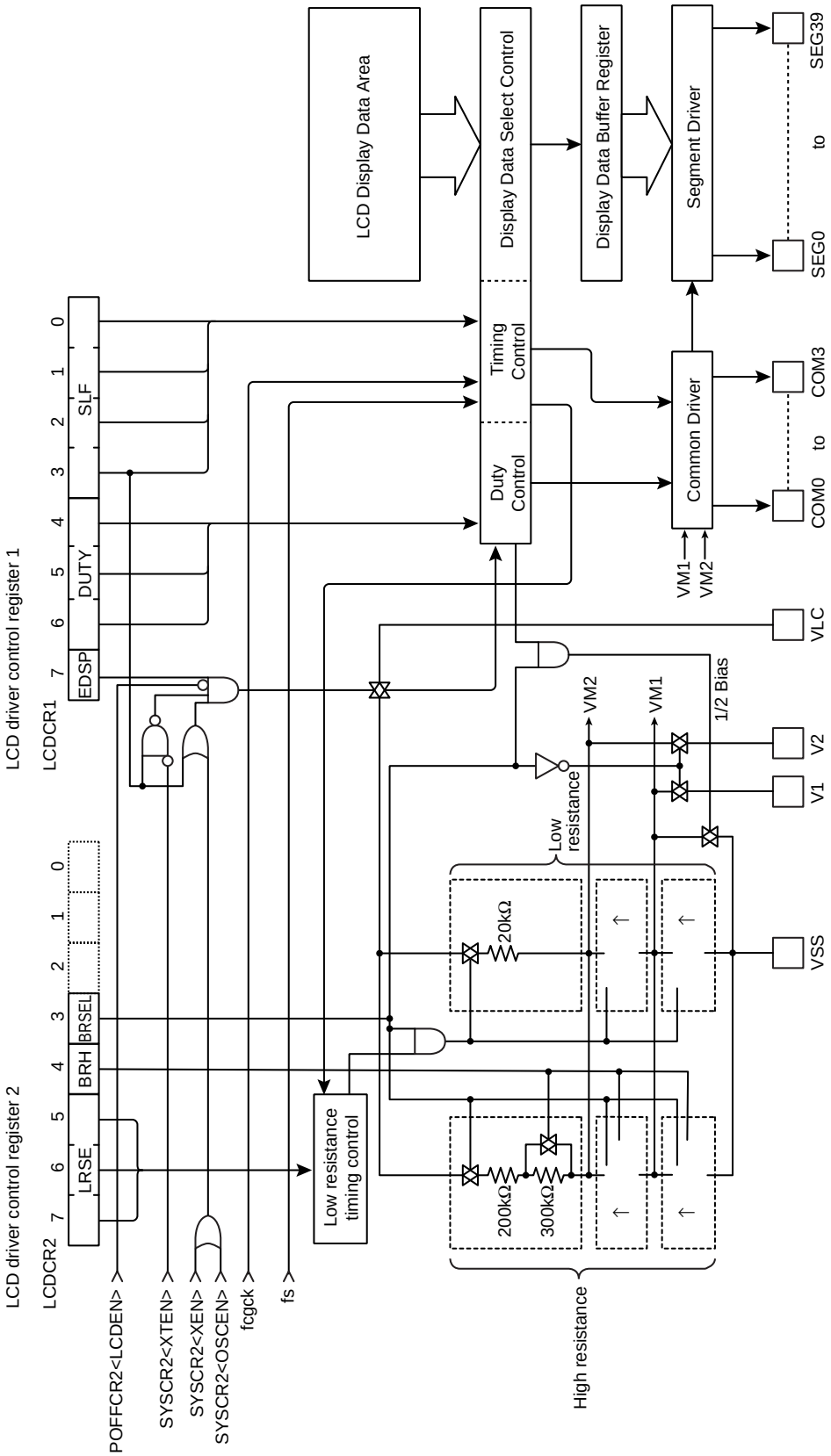


Figure 23-1 LCD Driver

Downloaded from Arrow.com.

23.2 Control

The LCD driver is controlled by the low power consumption register (POFFCR2), LCD control register 1 (LCDCR1), and LCD control register 2 (LCDCR2).

Low power consumption register 2

POFFCR2		7	6	5	4	3	2	1	0
(0x00F76)	Bit Symbol	LCDEN	-	RTCEN	-	-	-	SIO1EN	SIO0EN
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0

LCDEN	LCD Control	0 1	Disable Enable
RTCEN	RTC Control	0 1	Disable Enable
SIO1EN	SIO1 Control	0 1	Disable Enable
SIO0EN	SIO0 Control	0 1	Disable Enable

Downloaded from Arrow.com.

LCD driver control register 1

LCDCR1		7	6	5	4	3	2	1	0
(0x00E7C)	Bit Symbol	EDSP	DUTY			SLF			
	Read/Write	R/W	R/W			R/W			
	After reset	0	0	0	0	0	0	0	0

EDSP	LCD display control	0	Blank LCD display
		1	Enable LCD display
DUTY	LCD drive method selection	000	1/4 duty (1/3 bias)
		001	1/3 duty (1/3 bias)
		010	1/3 duty (1/2 bias)
		011	1/2 duty (1/2 bias)
		100	Static
		101	Reserved
		110	Reserved
		111	Reserved
SLF	Base frequency selection	0000	$fcgck/2^{18}$
		0001	$fcgck/2^{17}$
		0010	$fcgck/2^{16}$
		0011	$fcgck/2^{15}$
		0100	$fcgck/2^{14}$
		0101	$fcgck/2^{13}$
		0110	$fcgck/2^{12}$
		0111	Reserved
		1000	$fs/2^{9}$
		1001	$fs/2^{8}$
		1010	Reserved
		to	
		1111	Reserved

Note 1: fcgck = Gear clock [Hz], fs = Low-frequency clock [Hz]

Note 2: In SLOW2 mode, do not set SLF to 0y0000 to 0y0110 (i.e. frequencies based on fcgck). If SLF is set to one of these frequencies, pulses of an unexpected frame frequency will be output from the common and segment output pins.

Downloaded from Arrow.com.

LCD driver control register 2

LCDCR2		7	6	5	4	3	2	1	0
(0x00E7D)	Bit Symbol	LRSE			BRH	BRSEL			
	Read/Write	R/W			R/W	R/W	R	R	R
	After reset	0	0	0	0	0	0	0	0

			LCDCR1<SLF> setting								
			0000	0001	0010	0011	0100	0101	0110	1000	1001
LRSE	Low internal bleeder resistance connection time selection	000	Not connected								
		001	2^{11}/fcgck	2^{10}/fcgck	2^{9}/fcgck	2^{8}/fcgck	2^{7}/fcgck	2^{6}/fcgck	2^{5}/fcgck	2^{2}/fs	2/fs
		010	2^{12}/fcgck	2^{11}/fcgck	2^{10}/fcgck	2^{9}/fcgck	2^{8}/fcgck	2^{7}/fcgck	2^{6}/fcgck	2^{3}/fs	2^{2}/fs
		011	2^{13}/fcgck	2^{12}/fcgck	2^{11}/fcgck	2^{10}/fcgck	2^{9}/fcgck	2^{8}/fcgck	2^{7}/fcgck	2^{4}/fs	2^{3}/fs
		100	2^{14}/fcgck	2^{13}/fcgck	2^{12}/fcgck	2^{11}/fcgck	2^{10}/fcgck	2^{9}/fcgck	2^{8}/fcgck	2^{5}/fs	2^{4}/fs
		101	2^{15}/fcgck	2^{14}/fcgck	2^{13}/fcgck	2^{12}/fcgck	2^{11}/fcgck	2^{10}/fcgck	2^{9}/fcgck	2^{6}/fs	2^{5}/fs
		110	Always connected								
		111	Reserved								
BRH	High internal bleeder resistance selection	0	200 kΩ (Typ.)								
		1	500 kΩ (Typ.)								
BRSEL	Internal/external bleeder resistance switching control	0	Use external bleeder resistance (Note 4)								
		1	Use internal bleeder resistance								

Note 1: fcgck = Gear clock [Hz], fs = Low-frequency clock [Hz]

Note 2: The LRSE and BRH settings are effective only when BRSEL is set to select internal bleeder resistance.

Note 3: When a read instruction is executed on LCDCR2, bits 2 to 0 are read as "0".

Note 4: The external bleeder resistance connection pins are shared with the segment output pins SEG38,39. When external bleeder resistance is used, SEG38,39 cannot be used for segment output.

Downloaded from Arrow.com.

23.3 Low Power Consumption Function

The LCD driver has the low power consumption register (POFFCR2) to reduce unnecessary power consumption when the display function is not used.

Setting POFFCR2<LCDEN> to "0" disables the basic clock supply to the LCD driver to reduce unnecessary power consumption. The LCD driver cannot be used in this state. Setting POFFCR2 <LCDEN> to "1" enables the basic clock supply to the LCD driver, allowing the LCD driver to be used.

After a reset, POFFCR2 <LCDEN> is initialized to "0", disabling the LCD driver. Therefore, to use the LCD driver, be sure to set POFFCR2 <LCDEN> to "1" in the initialization part of the program (before setting the LCD driver control registers).

Do not change POFFCR2<LCDEN> to "0" while the LCD driver is operating. Doing so may cause the LCD driver to behave in an unexpected manner.

Downloaded from Arrow.com.

23.4 Functions

23.4.1 LCD Display Control (LCDCR1<EDSP>)

When LCDCR1<EDSP> is set to "1", the power switch of the LCD driver is turned on and VLC voltage is applied to the LCD driver, enabling the display function. Setting LCDCR1<EDSP> to "0" turns off the power switch of the LCD driver and shuts off VLC voltage, blanking the display.

Table 23-1 shows the pin states when the display is enabled and when the display is blank.

Table 23-1 Pin States

PxLCR	LCDCR2 <BRSEL>	LCDCR1 <EDSP>	Common output pins	Multiplexed pins (Input/output port) (Segment output)	Multiplexed pins (Input/output port) (Segment output) (External bleeder resistance connection)
0	0	0	L level	Input/output port	External bleeder resistance connection
0	0	1	Common output		
0	1	0	L level		Input/output port
0	1	1	Common output		
1	0	0	L level	L level	External bleeder resistance connection
1	0	1	Common output	Segment output	
1	1	0	L level	L level	L level
1	1	1	Common output	Segment output	Segment output

Note:"x" denotes an I/O port number.

23.4.1.1 Operation at Reset

When a reset occurs, LCDCR1<EDSP> is initialized to "0" and the power switch of the LCD driver is automatically turned off, shutting off VLC voltage. At this time, the common output pins are fixed to the L level. The multiplexed pins (input/output port or segment output) are configured as port input pins (high impedance). Therefore, if external reset operation takes time, the LCD display may become blurred. The multiplexed pins (input/output port, external bleeder resistance connection or segment output) are configured as external bleeder resistance connection pins.

23.4.1.2 Operation in IDLE0, SLEEP0 and STOP modes

When IDLE0, SLEEP0 or STOP mode is activated while LCDCR1<EDSP> is set to "1", LCDCR1 <EDSP> is automatically initialized to "0", blanking the LCD display. After IDLE0, SLEEP0 or STOP mode is exited, LCDCR1<EDSP> must be set to "1" to re-enable the LCD display.

23.4.1.3 Operation in SLOW mode

When the LCD is used in both NORMAL2 mode and SLOW1/2 mode, it is recommended that LCDCR1<SLF> be set to a frequency based on fs (0y1000 or 0y1001). (This will eliminate the need for changing the LCDCR1<SLF> setting each time the operating mode is switched between NORMAL2 mode and SLOW1/2 mode.)

If a frequency based on fcgck is used in NORMAL 2 mode, it is necessary to clear LCDCR1<EDSP> to "0" before switching to SLOW2 mode. Then, after entering SLOW2 mode, it is necessary to change LCDCR1<SLF> to a frequency based on fs and to set LCDCR1<EDSP> to "1". Likewise, in switching

Downloaded from Arrow.com.

from SLOW2 mode to NORMAL2 mode, it is necessary to clear LCDCR1<EDSP> to "0" before switching to NORMAL2 mode. Then, after entering NORMAL2 mode, it is necessary to change LCDCR1<SLF> to a frequency based on fcgck and to set LCDCR1<EDSP> to "1".

23.4.1.4 Display operation according to the base frequency setting (LCDCR1<SLF>) (fail-safe)

When LCDCR1<SLF> is set to 0y0000 to 0y0110, the high-frequency clock must be activated (SYSCR2<XEN>="1" or SYSCR2<OSCEN>="1") and allowed to achieve stable oscillation before LCDCR1<EDSP> can be set to "1". If LCDCR1<EDSP> is set to "1" while the high-frequency clock is stopped, the LCD display cannot be enabled. (Although LCDCR1<EDSP> changes to "1", the LCD display remains blank.)

Likewise, when LCDCR1<SLF> is set to 0y1000 to 0y1001, the low-frequency clock must be activated (SYSCR2<XTEN>="1") and allowed to achieve stable oscillation before LCDCR1<EDSP> can be set to "1". If LCDCR1<EDSP> is set to "1" while the low-frequency clock is stopped, the LCD display cannot be enabled. (Although LCDCR1<EDSP> changes to "1", the LCD display remains blank.)

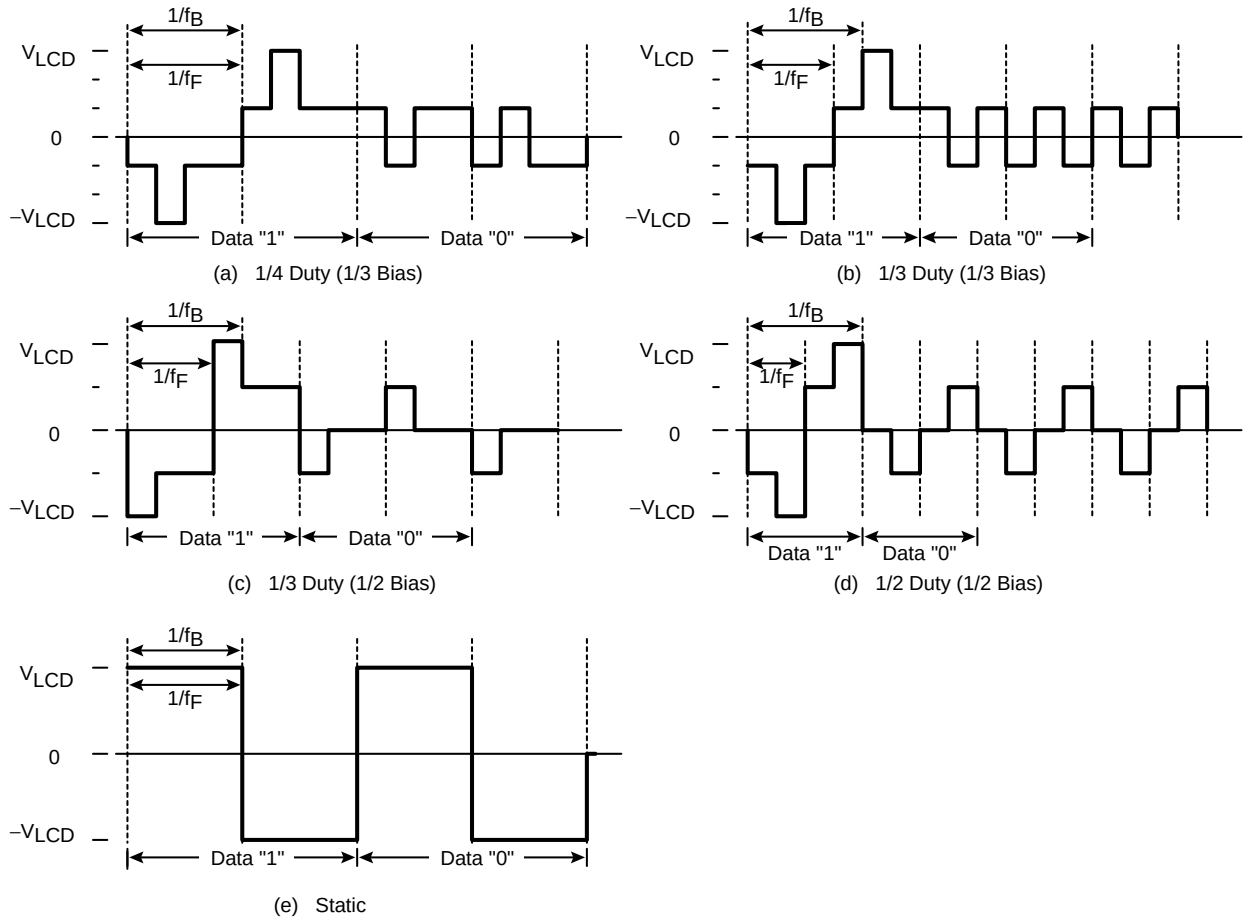
23.4.1.5 Display operation according to the low power consumption register (fail-safe)

When LCDCR1<EDSP>="1", setting POFFCR2<LCDEN> to "0" blanks the LCD display. The display can be re-enabled by setting POFFCR2<LCDEN> to "1".

Downloaded from Arrow.com.

23.4.2 LCD Drive Methods (LCDCR1<DUTY>)

The LCD drive method can be selected from the following five types by the setting of LCDCR1<DUTY>.



Note 1: f_F = Frame frequency

Note 2: f_B = Base frequency (LCDCR1<SLF>)

Note 3: V_{LCD3} = LCD drive voltage (= $V_{LC} - V_{SS}$)

Figure 23-2 LCD Drive Waveforms (Potential Difference between COM and SEG Pins)

Downloaded from Arrow.com.

23.4.3 Frame Frequency (LCDCR1<SLF>)

The frame frequency (f_F) is determined based on the drive method and base frequency, as shown in Table 23-2. The base frequency is selected by LCDCR1<SLF>.

Table 23-2 Frame Frequency Settings

SLF	Base frequency [Hz]	Frame frequency [Hz]			
		1/4 Duty	1/3 Duty	1/2 Duty	Static
0000	fcgck / 2^{18}	fcgck / 2^{18}	(4 / 3) x fcgck / 2^{18}	(4 / 2) x fcgck / 2^{18}	fcgck / 2^{18}
	(fcgck = 16 MHz)	61	81	122	61
0001	fcgck / 2^{17}	fcgck / 2^{17}	(4 / 3) x fcgck / 2^{17}	(4 / 2) x fcgck / 2^{17}	fcgck / 2^{17}
	(fcgck = 16 MHz)	122	163	244	122
	(fcgck = 8 MHz)	61	81	122	61
0010	fcgck / 2^{16}	fcgck / 2^{16}	(4 / 3) x fcgck / 2^{16}	(4 / 2) x fcgck / 2^{16}	fcgck / 2^{16}
	(fcgck = 8 MHz)	122	163	244	122
	(fcgck = 4 MHz)	61	81	122	61
0011	fcgck / 2^{15}	fcgck / 2^{15}	(4 / 3) x fcgck / 2^{15}	(4 / 2) x fcgck / 2^{15}	fcgck / 2^{15}
	(fcgck = 4 MHz)	122	163	244	122
	(fcgck = 2 MHz)	61	81	122	61
0100	fcgck / 2^{14}	fcgck / 2^{14}	(4 / 3) x fcgck / 2^{14}	(4 / 2) x fcgck / 2^{14}	fcgck / 2^{14}
	(fcgck = 2 MHz)	122	163	244	122
	(fcgck = 1 MHz)	61	81	122	61
0101	fcgck / 2^{13}	fcgck / 2^{13}	(4 / 3) x fcgck / 2^{13}	(4 / 2) x fcgck / 2^{13}	fcgck / 2^{13}
	(fcgck = 1 MHz)	122	163	244	122
	(fcgck = 0.5 MHz)	61	81	122	61
0110	fcgck / 2^{12}	fcgck / 2^{12}	(4 / 3) x fcgck / 2^{12}	(4 / 2) x fcgck / 2^{12}	fcgck / 2^{12}
	(fcgck = 0.5 MHz)	122	163	244	122
	(fcgck = 0.25 MHz)	61	81	122	61
1000	fs / 2^9	fs / 2^9	(4 / 3) x fs / 2^9	(4 / 2) x fs / 2^9	fs / 2^9
	(fs = 32.768 kHz)	64	85	128	64
1001	fs / 2^8	fs / 2^8	(4 / 3) x fs / 2^8	(4 / 2) x fs / 2^8	fs / 2^8
	(fs = 32.768 kHz)	128	171	256	128

Note: fcgck = Gear clock frequency [Hz], fs = Low-frequency clock frequency [Hz]

Downloaded from Arrow.com.

23.4.4 Internal/External Bleeder Resistance Switching Control

The LCD bias voltage is generated by bleeder resistance. Either external or internal bleeder resistance can be used.

To use internal bleeder resistance, set LCDCR2<BRSEL> to "1". In this case, the multiplexed pins (input/output port, external bleeder resistance connection or segment output) can be configured as input/output ports or segment output pins.

To use external bleeder resistance, set LCDCR2<BRSEL> to "0" and connect external resistance to the external bleeder resistance connection pins (V1, V2). In this case, the multiplexed pins (input/output port, external bleeder resistance connection, or segment output) can only be used as external bleeder resistance connection pins.

See Figure 23-4 for how to connect bleeder resistance.

23.4.5 Low Internal Bleeder Resistance Connection Time Selection (LCDCR2<LRSE>)

Internal bleeder resistance is comprised of two parts: high resistance and low resistance. The high and low resistance parts are connected in parallel for each bias voltage. The low bleeder resistance is provided with an analog switch, and the time to turn on the low bleeder resistance can be adjusted by LCDCR2<LRSE>. While the analog switch is turned on, the low resistance is connected in parallel to the high resistance. This reduces the total amount of resistance, allowing the drive capability of the LCD driver to be increased.

Typically, the longer the period of connecting the low resistance, the higher the drive capability of the LCD panel, but the higher the power consumption. Conversely, the shorter the period of connecting the low resistance, the lower the drive capability, but the lower the power consumption. Insufficient drive capability will cause adverse effects on the LCD display, such as blurring. Choose the optimum drive capability for the LCD panel to be used.

Table 23-3 shows the connection time (percentage) of the low bleeder resistance per frame and the estimated amount of current that flows through the entire bleeder resistance in each case.

Figure 23-3 shows the bleeder resistance control timing for the 1/4 duty and 1/3 bias LCD.

See "23.4.6 High Internal Bleeder Resistance Selection (LCDCR2<BRH>)"for the setting of LCDCR2<BRH>.

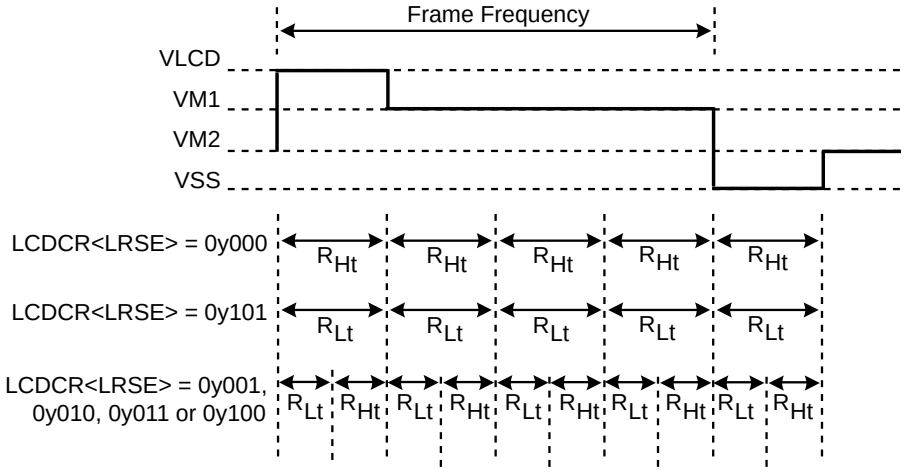
Downloaded from Arrow.com.

Table 23-3 Low Bleeder Resistance Connection Time (%) and Total Bleeder Current Values (Estimated)

LCDCR2 <LRSE>			1/4 Duty (1/3 Bias)		1/3 Duty (1/3 Bias)		1/3 Duty (1/2 Bias)		1/2 Duty (1/2 Bias)	
			BRH="1"	BRH="0"	BRH="1"	BRH="0"	BRH="1"	BRH="0"	BRH="1"	BRH="0"
000	Low resistance connection time (%)		0% (Always high resistance)							
	Bleeder current	VLC=5V	3.33μA	8.33μA	3.33μA	8.33μA	5.00μA	12.5μA	5.00μA	12.5μA
		VLC=3V	2.00μA	5.00μA	2.00μA	5.00μA	3.00μA	7.50μA	3.00μA	7.50μA
001	Low resistance connection time (%)		3.13%							
	Bleeder current	VLC=5V	5.94μA	10.94μA	5.94μA	10.94μA	8.91μA	16.41μA	8.91μA	16.41μA
		VLC=3V	3.56μA	6.56μA	3.56μA	6.56μA	5.34μA	9.84μA	5.34μA	9.84μA
010	Low resistance connection time (%)		6.25%							
	Bleeder current	VLC=5V	8.54μA	13.54μA	8.54μA	13.54μA	12.81μA	20.31μA	12.81μA	20.31μA
		VLC=3V	5.13μA	8.13μA	5.13μA	8.13μA	7.69μA	12.19μA	7.69μA	12.19μA
011	Low resistance connection time (%)		12.5%							
	Bleeder current	VLC=5V	13.75μA	18.75μA	13.75μA	18.75μA	20.63μA	28.13μA	20.63μA	28.13μA
		VLC=3V	8.25μA	11.25μA	8.25μA	11.25μA	12.38μA	16.88μA	12.38μA	16.88μA
100	Low resistance connection time (%)		25%							
	Bleeder current	VLC=5V	24.17μA	29.17μA	24.17μA	29.17μA	36.25μA	43.75μA	36.25μA	43.75μA
		VLC=3V	14.50μA	17.50μA	14.50μA	17.50μA	21.75μA	26.25μA	21.75μA	26.25μA
101	Low resistance connection time (%)		50%							
	Bleeder current	VLC=5V	45.00μA	50.00μA	45.00μA	50.00μA	67.50μA	75.00μA	67.50μA	75.00μA
		VLC=3V	27.00μA	30.00μA	27.00μA	30.00μA	40.50μA	45.00μA	40.50μA	45.00μA
110	Low resistance connection time (%)		100% (Always connected)							
	Bleeder current	VLC=5V	86.67μA	91.67μA	86.67μA	91.67μA	130.0μA	137.5μA	130.0μA	137.5μA
		VLC=3V	52.00μA	55.00μA	52.00μA	55.00μA	78.00μA	82.50μA	78.00μA	82.50μA

Note: The bleeder resistance current values shown above are estimated values. The actual current values may vary depending on the amount of LCD load and manufacturing variations in resistance values.

Downloaded from Arrow.com.



R_{Lt} : Period during which low resistance is connected
(high resistance and low resistance are connected in parallel)

R_{Ht} : Period during which low resistance is not connected
(only high resistance is connected)

Figure 23-3 Bleeder Resistance Selection by LCDCR2<LRSE> (1/4 Duty, 1/3 Bias)

Downloaded from Arrow.com.

23.4.6 High Internal Bleeder Resistance Selection (LCDCR2<BRH>)

The value of high internal bleeder resistance can be selected from two levels (500 kΩ (Typ.) or 200 kΩ (Typ.)) by the setting of LCDCR2<BRH>. Typically, the lower the resistance value, the higher the drive capability of the LCD panel, but the higher the power consumption. Conversely, the higher the resistance value, the lower the drive capability, but the lower the power consumption.

The resistance value of the low resistance is fixed to 20 kΩ (Typ.). Since the low resistance is connected in parallel to the high resistance via an analog switch, the total amount of resistance can be adjusted by the setting of LCDCR2<LRSE> as shown in Table 23-4.

For example, setting LCDCR2<BRH> to "1" selects a synthesized resistance of 19.23 kΩ (Typ.) when the low resistance is connected, and a high resistance of 500 kΩ (Typ.) when the low resistance is not connected.

Table 23-4 Bleeder Resistance Values

LCDCR2<BRH>	When low resistance is not connected	When low resistance is connected
1	500 kΩ (Typ.)	19.23 kΩ (Typ.)
0	200 kΩ (Typ.)	18.18 kΩ (Typ.)

Downloaded from Arrow.com.

23.5 LCD Display Operation

The LCD drive voltage V_{LCD} is given by the potential difference between the VLC and VSS pins ($V_{LC} - V_{SS}$). The LCD lights up when the potential difference between segment and common outputs is $\pm VL_{CD}$. At other times, the LCD is turned off.

Power supply connections should be made to satisfy the condition VLC ≤ VDD. Connection examples are shown in Figure 23-4.

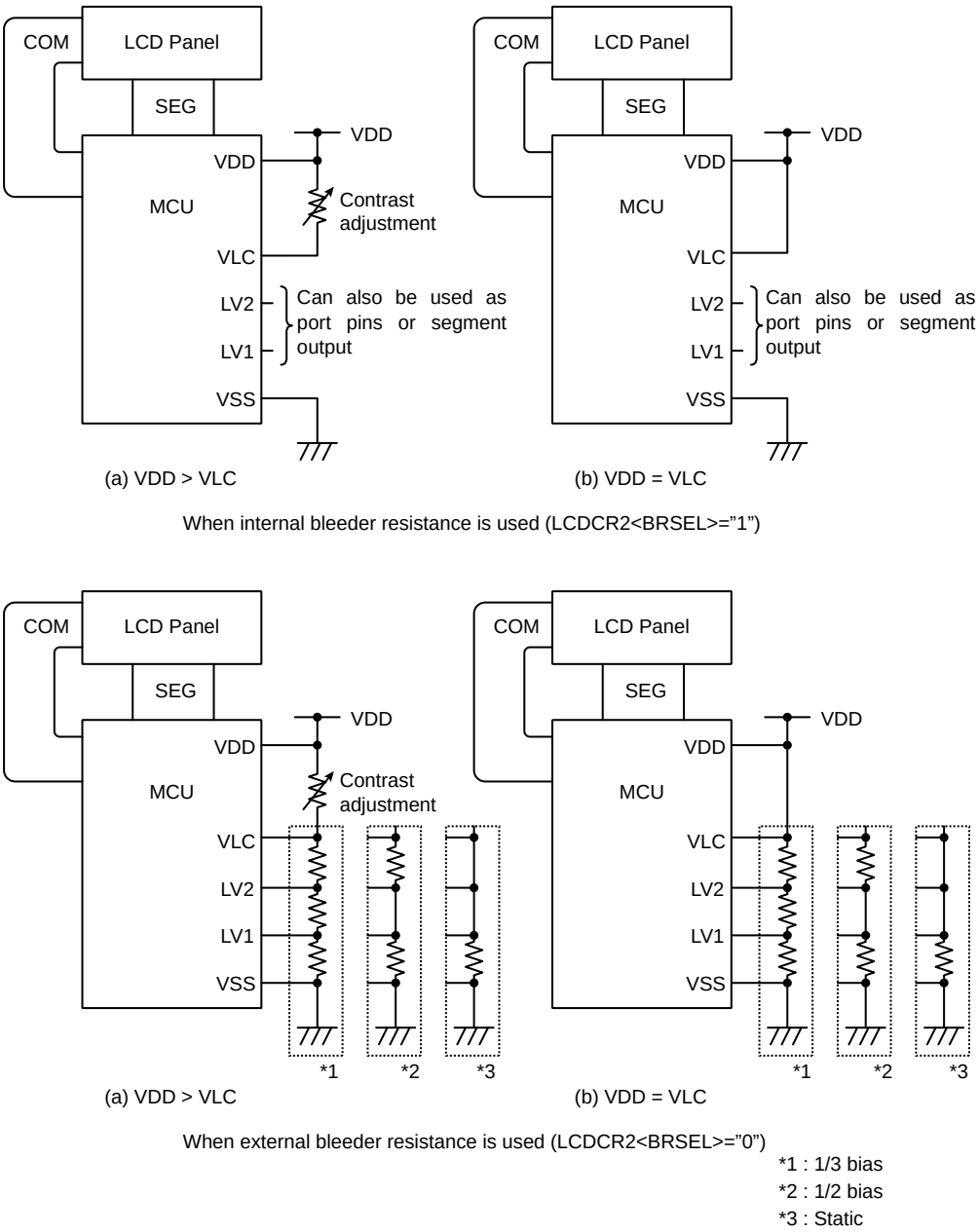


Figure 23-4 Connection Examples

Note 1: When the CPU operating voltage is the same as the LCD drive voltage, the VLC pin should be connected to the VDD pin.

Note 2: At reset, the common output pins become low. However, the multiplexed pins (input/output port or segment output) are configured as port input pins (high impedance). Therefore, if the multiplexed pins (input/output port or segment output) are used as segment output pins and an external reset signal is input for a prolonged period of time, the LCD display may be adversely affected, such as blurring. The multiplexed pins (input/output port, external bleeder resistance connection or segment output) are configured as external bleeder resistance connection pins.

Downloaded from Arrow.com.

23.6 Display Data Setting

Display data is stored in the display data area (20 bytes at addresses 0x00E40 to 0x00E53).

The display data stored in the display data area is automatically read out and sent to the LCD driver by hardware. The LCD driver generates the segment and common signals according to the display data and drive method. Therefore, display patterns can be changed by simply overwriting the contents of the display data area.Table 23-6 shows the correspondence between the display data area and the SEG and COM pins.

The LCD lights up when display data is "1" and is turned off when display data is "0". At reset, the data in the display data area (20 bytes at addresses 0x00E40 to 0x00E53) are initialized to "0".

Since the number of pixels that can be driven varies with the LCD drive method, the number of bits used for storing display data also varies accordingly. Therefore, the bits not used for storing display data and the data memory locations corresponding to addresses not connected to the LCD can be used for storing general user data (see Table 23-5).

Table 23-5 Bits To Be Used for Storing Display Data

Drive method	Bits 7/3	Bits 6/2	Bits 5/1	Bits 4/0
1/4 duty	COM3	COM2	COM1	COM0
1/3 duty	-	COM2	COM1	COM0
1/2 duty	-	-	COM1	COM0
Static	-	-	-	COM0

Note:" - " denotes bits not used for storing display data.

Table 23-6 LCD Display Data Area

Registar name (Address)	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/ Write	Initial value
LCDBUF00 (0x00E40)	SEG1				SEG0				R/W	(0000 0000)
LCDBUF01 (0x00E41)	SEG3				SEG2				R/W	(0000 0000)
LCDBUF02 (0x00E42)	SEG5				SEG4				R/W	(0000 0000)
LCDBUF03 (0x00E43)	SEG7				SEG6				R/W	(0000 0000)
LCDBUF04 (0x00E44)	SEG9				SEG8				R/W	(0000 0000)
LCDBUF05 (0x00E45)	SEG11				SEG10				R/W	(0000 0000)
LCDBUF06 (0x00E46)	SEG13				SEG12				R/W	(0000 0000)
LCDBUF07 (0x00E47)	SEG15				SEG14				R/W	(0000 0000)
LCDBUF08 (0x00E48)	SEG17				SEG16				R/W	(0000 0000)
LCDBUF09 (0x00E49)	SEG19				SEG18				R/W	(0000 0000)
LCDBUF10 (0x00E4A)	SEG21				SEG20				R/W	(0000 0000)
LCDBUF11 (0x00E4B)	SEG23				SEG22				R/W	(0000 0000)
LCDBUF12 (0x00E4C)	SEG25				SEG24				R/W	(0000 0000)
LCDBUF13 (0x00E4D)	SEG27				SEG26				R/W	(0000 0000)
LCDBUF14 (0x00E4E)	SEG29				SEG28				R/W	(0000 0000)
LCDBUF15 (0x00E4F)	SEG31				SEG30				R/W	(0000 0000)
LCDBUF16 (0x00E50)	SEG33				SEG32				R/W	(0000 0000)
LCDBUF17 (0x00E51)	SEG35				SEG34				R/W	(0000 0000)
LCDBUF18 (0x00E52)	SEG37				SEG36				R/W	(0000 0000)
LCDBUF19 (0x00E53)	SEG39				SEG38				R/W	(0000 0000)
	COM3	COM2	COM1	COM0	COM3	COM2	COM1	COM0		

Downloaded from Arrow.com.

23.7 Examples of How To Control the LCD Driver

23.7.1 Initialization

Figure 23-5 is a flowchart showing the initialization process of the LCD driver.

Example: When the LCD driver is to be operated with the following conditions:
- Drive method: 1/4 duty, 1/3 bias
- LCD frame frequency: fcgck/2^{18} [Hz]
- Connection time of low bleeder resistance (internal): 2^{15}/fcgck
- High bleeder resistance (internal): 200 kΩ

```
LD      (POFFCR2),0x80      ; LCDEN = 1
LD      (LCDCR1),0x00       ; set the LCD drive method and base frequency
LD      (LCDCR2),0x28       ; set the connection time of low bleeder resistance and the
                            ; high bleeder resistance value
LD      (PxLCR),0xFF        ; set the PxLCR register (x: I/O port number)
:       :
:       :                   ; set initial display data
SET     (LCDCR1).7          ; enable the LCD display
```

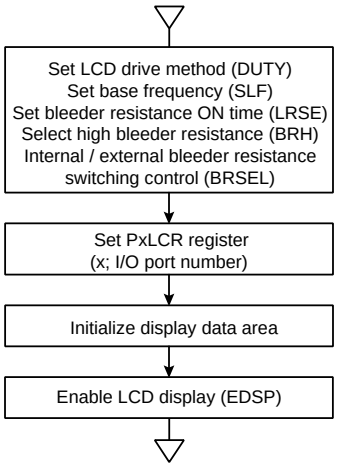


Figure 23-5 Flowchart for the LCD Driver Initialization

23.7.2 Display Data Setting

Display data is normally prepared as fixed data in the program memory (ROM) and transferred to the display data area by instructions.

Example: The following shows an example of how to set display data for displaying a number corresponding to BCD data stored at address 0x90 in the data memory by using the 1/4 duty and 1/3 bias LCD. Figure 23-6 shows an example of how the COM and SEG pins are connected to the LCD, and Table 23-7 shows how display data is set for this example.

Downloaded from Arrow.com.

```
LCDctl section code abs = 0x1F000
sLCDmain:     LD      C,(0x90)
              LD      HL,sTABLE
              LD      A,(HL+C)
              LD      (0x0E40),A
              RET
LCDdata section romdata abs = 0xF800
sTABLE:       DB      0y11011111, 0y00000110,
                      0y11100011, 0y10100111,
                      0y00110110, 0y10110101,
                      0y11110101, 0y00010111,
                      0y11110111, 0y10110111
```

Note:"DB" denotes a byte data definition instruction.

Figure 23-6 Example of COM and SEG Pin Connections (1/4 Duty)

Table 23-7 Example of Display Data (1/4 Duty)

No.	Display	Display data	No.	Display	Display data
0	0.	11011111	5	5	10110101
1	1	00000110	6	6	11110101
2	2	11100011	7	7	00000111
3	3	10100111	8	8	11110111
4	4	00110110	9	9	10110111

Downloaded from Arrow.com.

Example: The following shows an example of how to set display data for displaying a number as explained in example 1 by using the 1/2 duty LCD. Figure 23-7 shows an example of how the SEG and COM pins are connected to the LCD, and Table 23-8 shows how display data is set for this example.

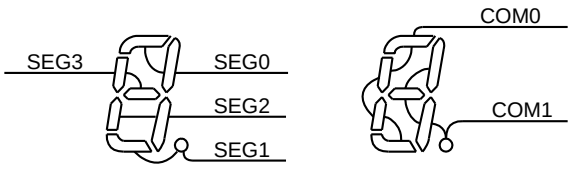


Figure 23-7 Example of COM and SEG Pin Connections

Table 23-8 Example of Display Data (1/2 Duty)

Number	Display data		Number	Display data	
	High-order address	Low-order address		High-order address	Low-order address
0	**01**11	**01**11	5	**11**10	**01**01
1	**00**10	**00**10	6	**11**11	**01**01
2	**10**01	**01**11	7	**01**10	**00**11
3	**10**10	**01**11	8	**11**11	**01**11
4	**11**10	**00**10	9	**11**10	**01**11

Note: An asterisk (*) denotes "don't care".

Downloaded from Arrow.com.

23.7.3 Drive Output Examples

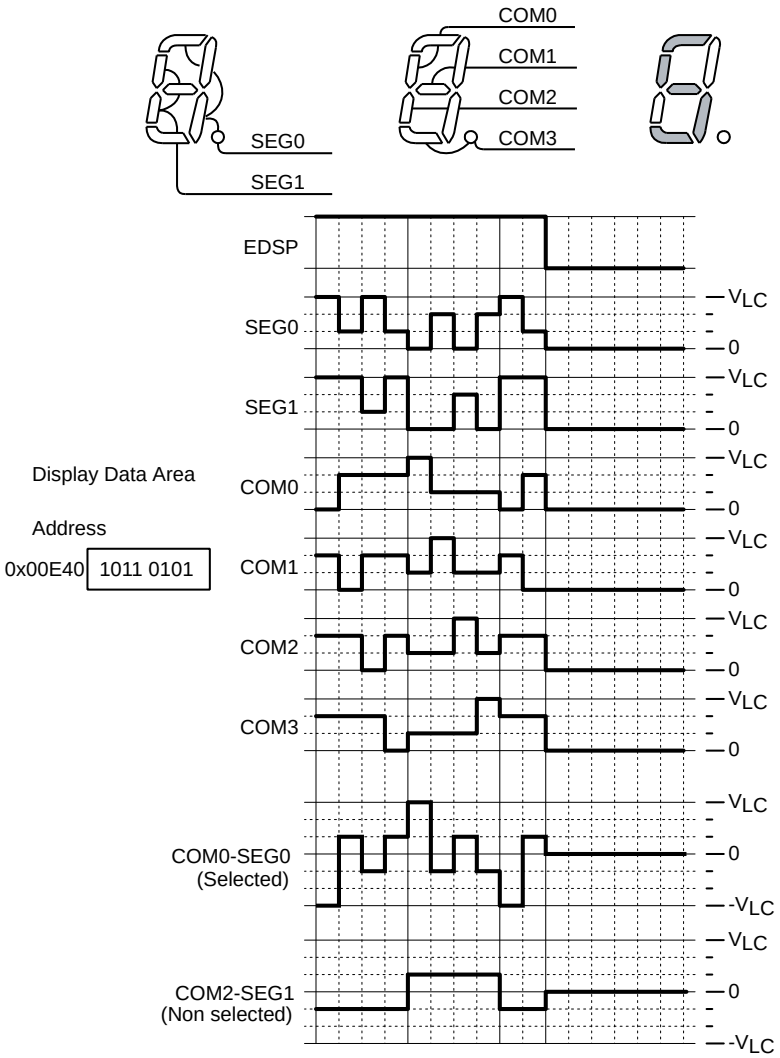


Figure 23-8 1/4 Duty (1/3 Bias) Drive

Downloaded from Arrow.com.

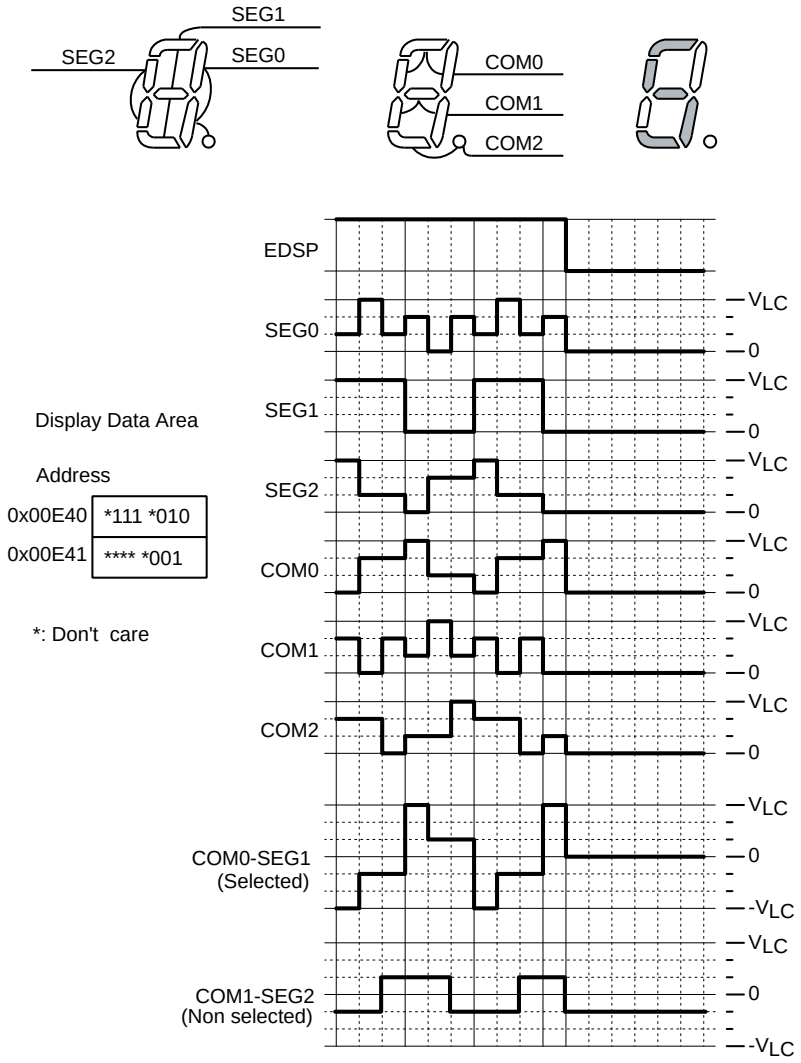


Figure 23-9 1/3 Duty (1/3 Bias) Drive

Downloaded from Arrow.com.

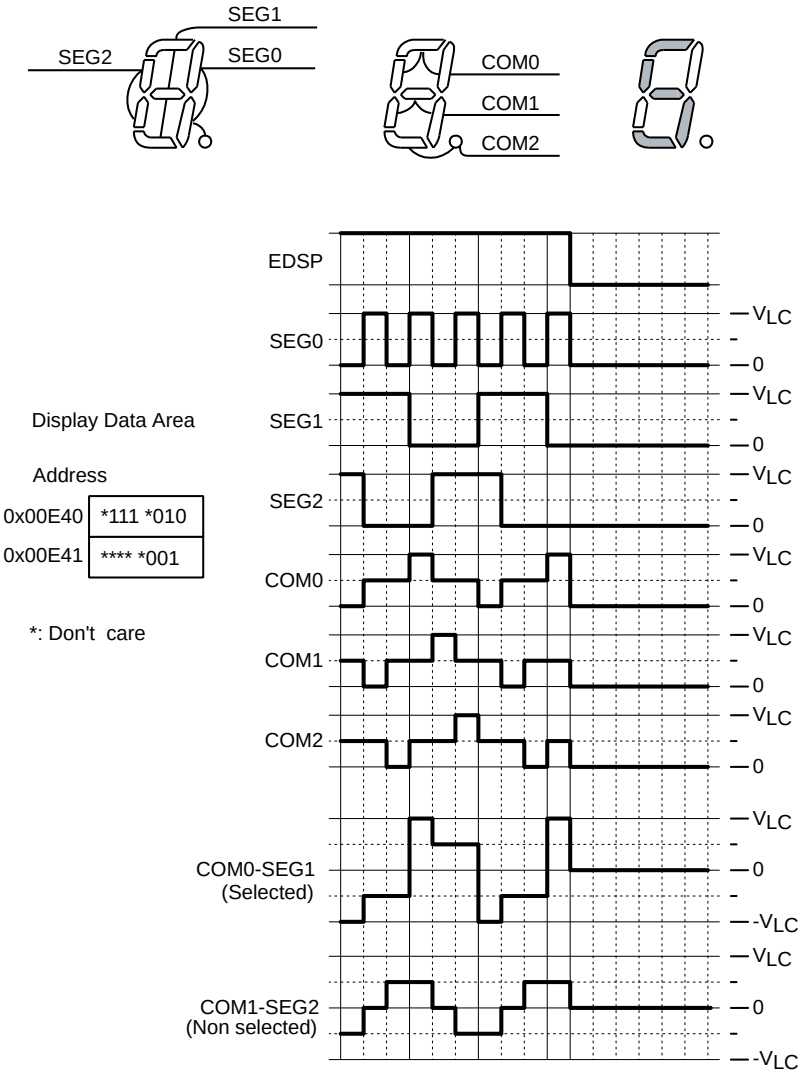


Figure 23-10 1/3 Duty (1/2 Bias) Drive

Downloaded from Arrow.com.

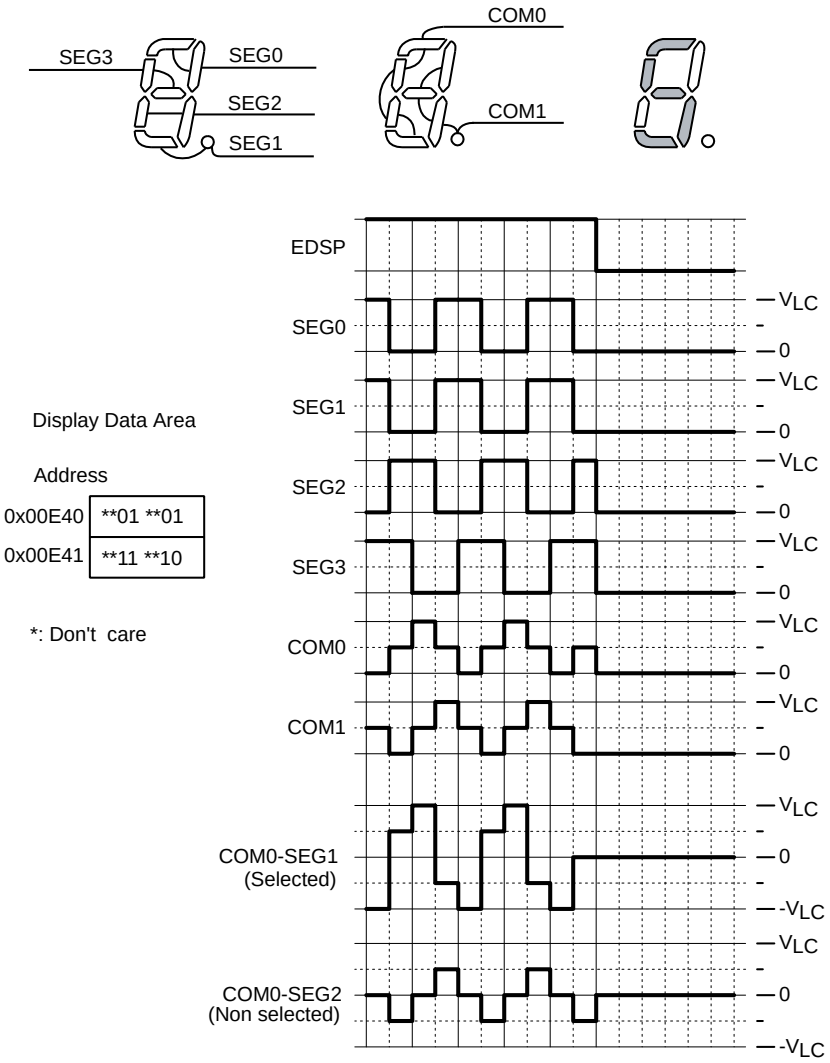


Figure 23-11 1/2 Duty (1/2 Bias) Drive

Downloaded from Arrow.com.

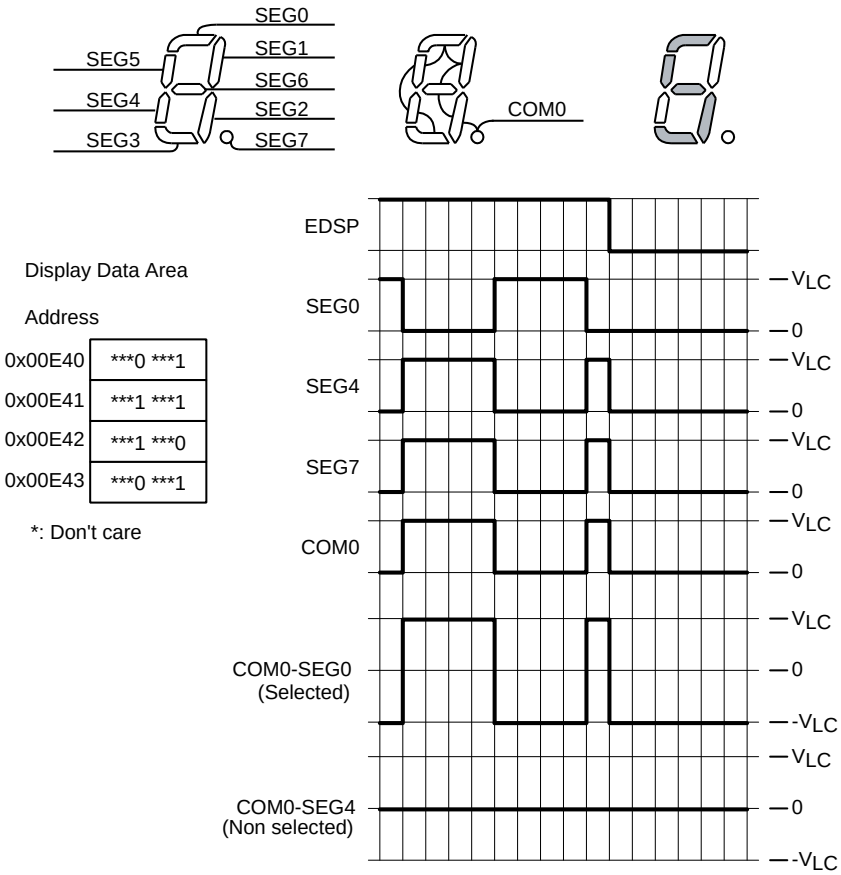


Figure 23-12 Static Drive

Downloaded from Arrow.com.

24. Flash Memory

The TMP89FW24A has flash memory of 126976 bytes. A write and erase operation to be performed on flash memory can be controlled in the following three modes:

- MCU mode

  In MCU mode, the flash memory is accessesd by the CPU control, and the flash memory can be executed the erasing and writing without affecting the operation of a running application. Therefore, this mode is used for software debugging and a firmware change after shipment of TMP89FW24A.

- Serial PROM mode

  In serial PROM mode, the flash memory is accesed by the CPU control. Use of the serial interface (UART and SIO) enables the flash memory to be controlled by the small number of pins. The TMP89FW24A used in serial PROM mode supports on-board programming, which enables users to program flash memory after the microcontroller is mounted on a user board.

- Parallel PROM mode

  The Parallel PROM mode allows the flash memory to be accessed as a stand-alone flash memory by the program writer provided by a third-party.High-speed access to the flash memory is available by controlling address and data signals directly. To receive a support serivice for the program writer, please ask a Toshiba sales representative.

In MCU mode and serial PROM modes, flash memory control registers (FLSCR1 and FLSCR2) are used to control the flash memory. This chapter describes how to access the flash memory using the MCU mode and serial PROM modes.

Following operations must be performed on the RAM or Shadow RAM.
- Performing write or erase operation on Flash memory.
- Setting or erasing the security.

Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

Note that in these processes interrupts must be prohibited; interrupt vectors and routines must be placed on RAM or Shadow RAM.

Applicable commands : Page Program, Chip Erase, Sector Erase, Security Program, Security Erase

When sector erasing, chip erasing, security setting or security releasing is performed by API on the BOOT ROM, hold wait time for 200μs or more before reading data from Flash memory or starting instruction fetch after the completion of calling API.

Note that calling API and holding wait time must be performed on RAM or Shadow RAM.

Applicable API : 0x1012(.BTEraseSec), 0x1014(.BTEraseChip), 0x1018(.BTSetSP), 0x1020(.BTErsSP)

Downloaded from Arrow.com.

24.1 Flash Memory Control

The flash memory is controlled by the flash memory register 1 (FLSCR1) and the flash memory control register 2 (FLSCR2).

Flash memory control register 1

FLSCR1 (0x00FD0)	7	6	5	4	3	2	1	0
Bit Symbol	FLSMD			BAREA	FAREA		"0"	"0"
Read/Write	R/W			R/W	R/W		R/W	R/W
After reset	0	1	0	0	0	0	0	0

FLSMD	Flash memory command sequence control	010: 101: Others:	Disable command sequence execution Enable command sequence execution Reserved	
BAREA	BOOTROM mapping control		MCU mode	Serial PROM mode
		0: 1:	Hide BOOTROM Show BOOTROM	- Show BOOTROM
FAREA	Flash memory area select control	00:	Assign the data area 0x08000 through 0x0FFFF to the data area 0x08000 through 0x0FFFF(standard mapping)	
		01:	Assign the data area 0x01000 through 0x07FFF to the data area 0x09000 through 0x0FFFF.	
		10:	Assign the data area 0x18000 through 0x1FFFF to the data area 0x08000 through 0x0FFFF.	
		11:	Assign the data area 0x10000 through 0x17FFF to the data area 0x08000 through 0x0FFFF.	

Note 1: Reserved : It is prohibited to make a setting in "Reserved".

Note 2: The flash memory control register 1 has a double-buffer structure comprised of the register FLSCR1 and a shift register. Writing "0xD5" to the register FLSCR2 allows a register setting for the register FLSCR1 to be reflected and take effect in the shift register. This means that a register setting value does not take effect until "0xD5" is written to the register FLSCR2. The value of the shift register can be checked by reading the register FLSCRM.

Note 3: FLSMD must be set either “0y010” or “0y101”.

Note 4: Make sure that you write "0" to the bits 0 and 1 of the register FLSCR1.

Flash memory control register 2

FLSCR2 (0x00FD1)	7	6	5	4	3	2	1	0
Bit Symbol	CR1EN							
Read/Write	W							
After reset	*	*	*	*	*	*	*	*

CR1EN	FLSCR1 register enable/disable control	0xD5 Others	Enable a change in the FLSCR1 setting Reserved

Note 1: If "0xD5" is set on FLSCR2<CR1EN> with FLSCR1<FLSMD> set to "101", the flash memory goes into an active state, and MCU consumes the same amount of current as it does during a read.

Downloaded from Arrow.com.

Flash memory status register

FLSCRM (0x00FD1)		7	6	5	4	3	2	1	0
	Bit Symbol	BUSY	-	FLSMDM	BAREAM	FAREAM		-	-
	Read/Write	R	R	R	R	R		R	R
	After reset	0	0	0	0	0	0	0	0

BUSY	Flash memory Ready / Busy status	0	-
		1	During write or erase operation of flash memory During write or erase operation of security bit
FLSMDM	Monitoring of FLSCR1<FLSMD> status	0	FLSCR1<FLSMD>="101" setting disabled
		1	FLSCR1<FLSMD>="101" setting enabled
BAREAM	Monitoring of FLSCR1<BAREA> status		Value of FLSCR1<BAREA> currenly enabled
FAREAM	Monitoring of FLSCR1<FAREA> status		Value of FLSCR1<FAREA> currently enabled

Note 1: Bits 5 through 2 of the FLSCRM is the register that checks the value of the shift regsiter of the flash memory control register 1.

Note 2: FLSMDM turns into "1" only if FLSMD="101" becomes effective.

Note 3: If an instruction to read FLSCRM is executed, "0" is read from the bit 6.

Note 4: In serial PROM mode, "1" is always read from BAREAM.

Note 5: Bits 1 and 0 of FLSCRM are read from the value of bits 1 and 0 of FLSCR1 at the time 0xD5 was set to FLSCR2.

Note 6: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

Port input control register (works only in serial PROM mode)

SPCR (0x00FD3)		7	6	5	4	3	2	1	0
	Bit Symbol	-	-	-	-	-	-	PIN1	PIN0
	Read/Write	R	R	R	R	R	R	R/W	R/W
	After reset	1	1	1	1	1	0	0	0

			In serial PROM mode	In MCU mode
PIN1	Port input control (SCLK0 pin) in serial PROM mode	0	Port input disabled	Input enabled for all ports. Nonfunctional whatever settings are made. "0" is read.
		1	Port input enabled	
PIN0	Port input control (except RXD0, TXD0 and SCLK0) in serial PROM mode	0	Port input disabled	
		1	Port input enabled	

Note 1: A read or a write operation can be performed on the SPCR register only in serial PROM mode. If a write operation is performed on this register in MCU mode, the port input control does not function. If a read operation is performed on the SPCR register in MCU mode, "0" is read from bits 7 through 0.

Note 2: All I/O ports are controlled by PIN0, except the ports RXD0, TXD0 and SCLK0 which are used in serial PROM mode. By using PIN1, the SCLK0 pin can be configured separately.

Downloaded from Arrow.com.

24.2 Functions

24.2.1 Flash memory command sequence execution (FLSCR1<FLSMD>)

To prevent inadvertent writes to the flash memory due to program error or microcontroller malfunction, the execution of the flash memory command sequence can be disabled (the flash memory can be write protected) by making an appropriate setting to the control register (write protection). To enable the execution of the command sequence, set FLSCR1<FLSMD> to "0y101", and then set FLSCR2<CR1EN> to "0xD5". To disable the execution of the command sequence, set FLSCR1<FLSMD> to "0y010", and then set FLSCR2<CR1EN> to "0xD5". If the command sequence is executed with the execution of the command sequence set to "disable", the executed command sequence takes no effect.

After a reset, FLSCR1<FLSMD> is initialized to "0y010" to disable the execution of the command sequence. FLSCR1<FLSMD> should normally set to "0y010" except when a write or a erase is to be performed on the flash memory.

Note 1: If FLSCR1<FLSMD> is set to "disable", subsequent commands (write instruction) generated are rejected but a command sequence being executed is not initialized.

If you want to set FLSCR1<FLSMD> to "disable", you must finish all command sequence and verify that the flash memory is ready to be read.

Downloaded from Arrow.com.

24.2.2 Flash memory area switching (FLSCR1<FAREA>)

To perform an erase or a write operation on the flash memory, a memory transfer instruction (command sequence) must be executed. If a memory transfer instruction is used to read or write data, a read or write operation can be performed only in the data area. To perform an erase or write operation on the code area, therefore, a part of the code area must be temporarily switched to the data area. This switching between data and code areas is performed by making the appropriate setting to FLSCR1 <FAREA>.

By setting "0xD5" on FLSCR<CR1EN> after setting FLSCR1<FAREA> to "01", 0x01000 through 0x07FFF (AREA D0) in the data area is mapped to 0x09000 through 0x0FFFF (AREA D1) in the data area.

By setting "0xD5" on FLSCR2<CR1EN> after setting <FAREA> to "10", 0x18000 through 0x1FFFF (AREA C1) in the code area is mapped to 0x08000 through 0x0FFFF (AREA D1).

By setting "0xD5" to FLSCR2<CR1EN> after setting FLSCR1<FAREA> to "11", 0x10000 through 0x17FFF (AREA C0) in the code area is mapped to 0x08000 through 0x0FFFF (AREA D1).

For example, to access 0x14000 in the code area, set "0xD5" on FLSCR<CR1EN> after setting FLSCR1<FAREA> to "11", and then execute the memory transfer instruction on 0xC000.

To restore the flash memory to the initial state of mapping, set FLSCR1<FAREA> to "00", and then set "0xD5" on FLSCR2<CR1EN>.

All flash memory areas can be accessed by performing the appropriate steps described above and then executing the memory transfer instruction on address 0x08000 through 0x0FFFF (AREA D1) in the data area.

Additionally, make sure not to access the area where the memory is not allocated using an instruction, and not to specify the area using JUMP or CALL instruction.

Downloaded from Arrow.com.

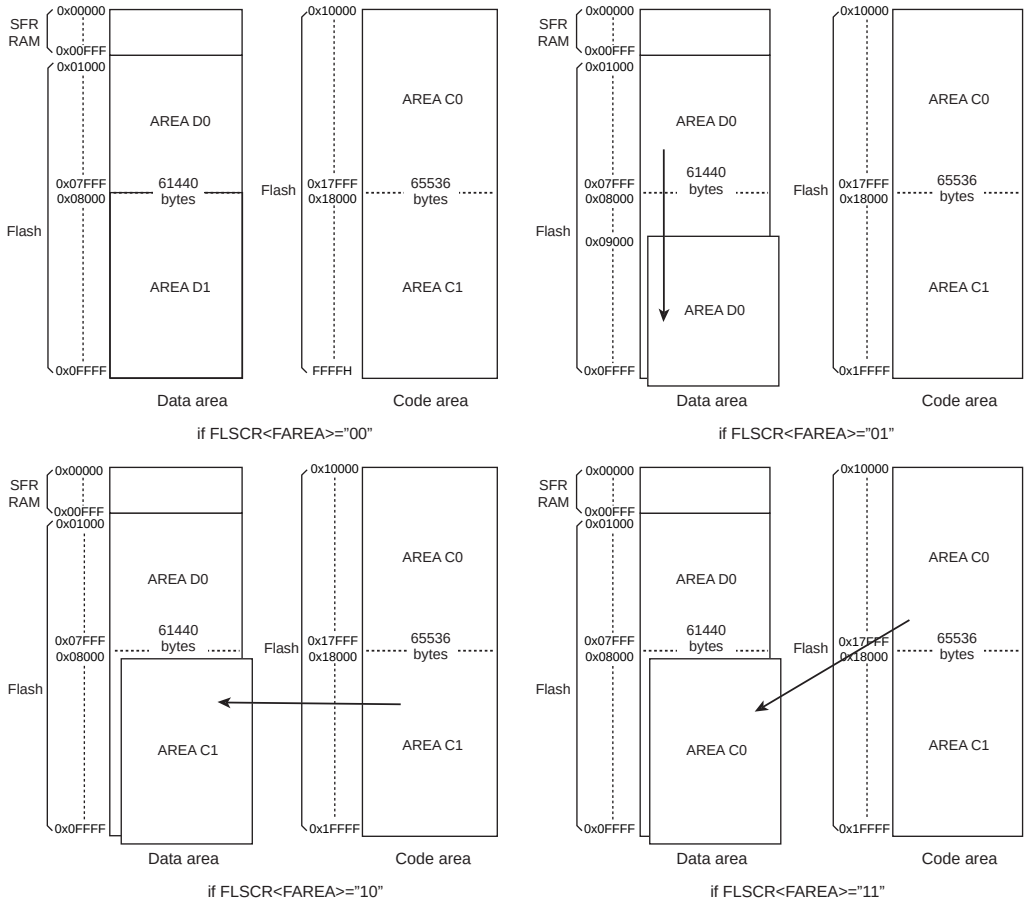


Figure 24-1 Area Switching Using the FLSCR1<FAREA> Setting

Downloaded from Arrow.com.

24.2.3 RAM area switching (SYSCR3<RAREA>)

If "0xD4" is set on SYSCR4 after setting SYSCR3<RAREA> to "1", RAM is mapped to the code area. To restore the RAM area to the initial state of mapping, set SYSCR3<RAREA> to "0", and then set "0xD4" on SYSCR4.

In serial PROM mode, RAM is mapped to the code area, irrespective of the SYSCR3<RAREA> setting.

Note: Do not allocate a program to switch SYSCR3<RAREA> to the code area from 0x0040 through 0x0FFF. If it is allocated to the area, the microcontroller may malfunction since the software does not function properly.

24.2.4 BOOTROM area switching (FLSCR1<BAREA>)

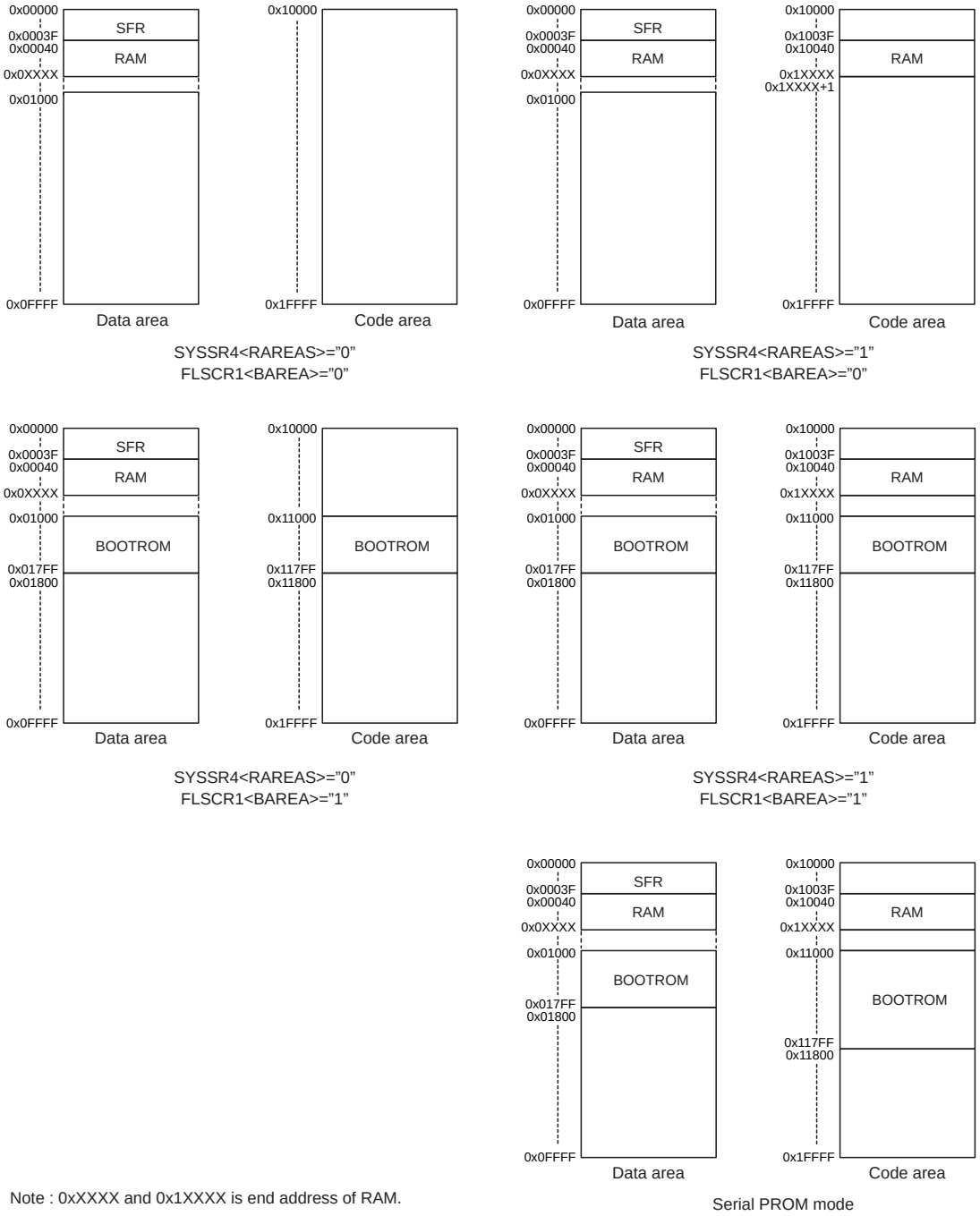
If "0xD5" is set to FLSCR2<CR1EN> after setting FLSCR1<BAREA> to "1" in MCU mode, the addresses from 0x1000 through 0x17FF in both data area and code area are masked by the flash memory, and 2K-byte (first half of 4KB) BOOTROM is mapped. If you do not want to map the BOOTROM, set "0" to FLSCR1<BAREA>, and then set "0xD5" on FLSCR2<CR1EN>.

BOOTROM stores program codes for programming the flash memory in serial PROM mode, and a part of BOOTROM area contains a support program (API) in order to write/erase the flash memory easily. Therefore, erase/write/read the flash memory is easily performed by calling the subroutine after mapping the BOOTROM.

In serial PROM mode, BOOTROM is mapped to the address from 0x1000 through 0x17FF in the data area and the address from 0x1000 through 0x1FFF in the code area, irrespective of the FLSCR1<BAREA> setting. BAREA is always "1", and the setting value of BAREA remains unchanged, even if you write data. "1" is always read from BAREA.

Note: Do not allocate a program to switch FLSCR1<FAREA> to the code area from 0x1000 through 0x1FFF. If it is allocated to the area, the microcontroller may malfunction since the software does not function properly.

Downloaded from Arrow.com.



Note : 0xXXXX and 0x1XXXX is end address of RAM.

Figure 24-2 Show/Hide Switching for BOOTROM and RAM

Downloaded from Arrow.com.

24.2.5 Monitoring a Ready / Busy state of the flash memory (FLSCRM<BUSY>)

Once command sequences (Page Program, Chip Erase, Sector Erase, Security Program and Security Erase) are executed, "1" is set to FLSCRM<BUSY> until the command sequences are completed. Completion of the execution can be checked using software. Normally, a read operation of FLSCRM<BUSY> is executed following the execution of these command sequences, and then perform polling until "0" is read from the flash memory.

Note: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

24.2.6 Port input control register (SPCR<PIN0,PIN1>)

In serial PROM mode, the input levels of all ports, except the ports RXD0 and TXD0 used in serial PROM mode, are physically fixed after a reset is released. This is designed to prevent a penetration current from flowing through unused ports (port inputs and functional peripheral inputs, which are also used as ports, are disabled). Therefore, to access the flash memory using the RAM loader mode and a method other than the UART, port inputs must be set to "enable". To enable the SCLK0 port input, set SPCR<PIN1> to "1". To enable port inputs other than RXD0, TXD0 and SCLK0 port inputs, set SPCR<PIN0> to "1".

In MCU mode, the SPCR register does not function.

24.2.7 Shadow RAM mapping control in the data area (SDWCR1<DADIS>)

In executing erase and write operations on the Flash memory (including API operations of the BOOT-ROM), command sequences or verify operations cannot be performed on addresses 0x0FC00 to 0x0FFFF when SDWCR1 <DADIS> is cleared to "0". Therefore, to erase or program the Flash memory, SDWCR1<DADIS> must be set to "1" to disable the mapping of shadow RAM in the data area

Downloaded from Arrow.com.

24.3 Command Sequence

In MCU mode and serial PROM mode, the command sequence consists of seven commands, as shown in Table 24-1.

Table 24-1 Command sequence

	Command sequence	1st Bus Write Cycle		2nd Bus Write Cycle		3rd Bus Write Cycle		4th Bus Write Cycle		5th Bus Write Cycle		6th Bus Write Cycle	
		Add	Data	Add	Data	Add	Data	Add	Data	Add	Data	Add	Data
1	Page Program	0x#555	0xAA	0x#AAA	0x55	0x#555	0xA0	PA (Note 1)	Data0 (Note 2)	PA (Note1)	Data1 (Note 2)	PA (Note 1)	Data2 (Note 2)
2	Sector Eraase (Partial Erase in Sector units)	0x#555	0xAA	0x#AAA	0x55	0x#555	0x80	0x#555	0xAA	0x#AAA	0x55	SA (Note 3)	0x30
3	Chip Erase (All Erase)	0x#555	0xAA	0x#AAA	0x55	0x#555	0x80	0x#555	0xAA	0x#AAA	0x55	0x#555	0x10
4	Product ID Entry	0x#555	0xAA	0x#AAA	0x55	0x#555	0x90	-	-	-	-	-	-
5	Product ID Exit	0x#XXX	0xF0	-	-	-	-	-	-	-	-	-	-
6	Security Program	0x#555	0xAA	0x#AAA	0x55	0x#555	0x9A	0x#555	0xAA	0x#AAA	0x55	0x#555	0x9A
7	Security Erase	0x#555	0xAA	0x#AAA	0x55	0x#555	0x6A	0x#555	0xAA	0x#AAA	0x55	0x#555	0x6A

	Command Sequence	7th Bus Write Cycle		8th Bus Write Cycle		9th Bus Write Cycle		- - -	130th Bus Write Cycle		131th Bus Write Cycle	
		Add	Data	Add	Data	Add	Data	- - -	Add	Data	Add	Data
1	Page Program	PA (Note 1)	Data3 (Note 2)	PA (Note 1)	Data4 (Note 2)	PA (Note 1)	Data5 (Note 2)	- - -	PA (Note 1)	Data 126 (Note 2)	PA (Note 1)	Data 127 (Note 2)
6	Security Program	SB (Note4)	0x9A	-	-	-	-	- - -	-	-	-	-
7	Security Erase	0x#XXX	0x6A	-	-	-	-	- - -	-	-	-	-

Note 1: PA : Page address

Specify the beginning address of the PA to be written. Refer to Table 24-3 about the addressable address. As the Page to be written is specified with the 4th address, addresses from 5th through 131st are not changed even if other Page is specified.

Note 2: Set the data in size of 128bytes (1 Page).

Note 3: SA : Sector address

Specify the Sector address to be erased. Refer to Table 24-5 for further details about the addressable address.

Note 4: SB : Security address

Specify the address of the Security bits. Refer to Table 24-6 for further details about the addressable address.

Note 5: Do not start the STOP, IDLE0/1/2 or SLEEP1/0 mode while a command sequence is being executed or a task specified in a command sequence is being executed (write, erase or ID Entry).

Note 6: # ; 0x1 through 0xF must be specified with the upper 4bits of the address. However, while FLSCRM<BAREAM> is set to "1", 0x2 or more must be specified. Usually, 0xF is recommended to be specified.

Note 7: Command sequences must be executed at 4 or more instruction cycles. If you use an instruction less than 4 instruction cycles, insert NOP instruction right after the write instruction to provide a instruction interval of 4 or more instruction cycle. A part of transfer instructions is shown in Table 24-2.

Note 8: The command sequence must be executed when the power supply source is turned ON (when SDWCR1<FLSOFF>="0" and SDWCR1<FLSWUE>="1"). Even if the command sequence is executed when the power supply of the flash memory is OFF, the command is not accepted.

Note 9: X ; Don't care

Downloaded from Arrow.com.

Table 24-2 Instruction Cycle of Transfer Instruction (Example)

Instruction cycle	Instruction
6	LD (vw), n
5	LD (vw), r
4	LD (DE), n
	LD (IX), n
	LD (IY), n
3	LD (HL), n
	LD (DE), r
	LD (HL), r (Note)
	LD (IX), r
	LD (IY), r

Note:LD (HL) and A are excepted (2 instruction cycle)

Downloaded from Arrow.com.

24.3.1 Page Program

This command sequence writes the flash memory in units of one Page. One command sequence can perform a write to one Page. One Page consists of 128bytes, 992 Page.The range of address that can be specified to the Page is shown in Table 24-4.

The address and the data to be written are specified in the Bus Write Cycle from 4th through 131th. The Bus Write Cycle from 4th through 131th (PA) must be specified as the beginning address of the Page to be written. When you set addresses using the instruction, refer to the Table 24-3 and set FLSCR1<FAREA> too. For example, to write data into the Page 1(0x01080 through 0x010FF) in the data area, set "0y01" to FLSCR1 <FAREA> , set "0xD5" to FLSCR2<CR1EN>, and then specify 0x9080 as the address in 4th through 131th address. The time needed to write each Page (the time between reception of the 131th command and the moment when the FLSCRM<BUSY> becomes "0") is 1.25ms (typ.). Other command sequences or an operation to read from the flash memory must not be executed until the write operation is completed. To check the completion of the write operation, perform a read operations on FLSCRM<BUSY>, and perform polling until "0" is read from the flash memory.

Note: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

Note 1: To rewrite data to address in the flash memory where data (including 0xFF) is already written, make sure that you erase the existing Page data by performing a Sector Erase or a Chip Erase before writing data.

Note 2: Do not perform a Page Program on the areas other than those shown in Table 24-3.

Note 3: Even if the PA is not the beginning address of the Page, a write is performed to the Page including the address.

Downloaded from Arrow.com.

Table 24-3 Address Range Specifiable (PA)

Write Area			FLSCR1 <FAREA>	Address from 4th through 131th (PA)
Sector	Page	Address		
AREA D0 (Data Area)	0	0x01000 through 0x0107F	01	0x9000
	1	0x01080 through 0x010FF		0x9080
	:	:		:
	222	0x07F00 through 0x07F7F		0xFF00
	223	0x07F80 through 0x07FFF		0xFF80
AREA D1 (Data Area)	224	0x08000 through 0x0807F	00	0x8000
	225	0x08080 thorugh 0x080FF		0x8080
	:	:		:
	478	0x0FF00 thorugh 0x0FF7F		0xFF00
	479	0x0FF80 through 0x0FFFF		0xFF80
AREA C0 (Code Area)	480	0x10000 through 0x1007F	11	0x8000
	481	0x10080 through 0x100FF		0x8080
	:	:		:
	734	0x17F00 through 0x17F7F		0xFF00
	735	0x17F80 through 0x17FFF		0xFF80
AREA C1 (Code Area)	736	0x18000 through 0x1807F	10	0x8000
	737	0x18080 through 0x180FF		0x8080
	:	:		:
	990	0x1FF00 through 0x1FF7F		0xFF00
	991	0x1FF80 through 0x1FFFF		0xFF80

Downloaded from Arrow.com.

Table 24-4 Page Configuration

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x01000	F															
0x01010																
0x01020																
0x01030								Page 0								
0x01040								(128 bytes)								
0x01050																
0x01060																
0x01070																E
0x01080	F															
0x01090																
0x010A0																
0x010B0								Page 1								
0x010C0								(128 bytes)								
0x010D0																
0x010E0																
0x010F0																E
0x1FF80	F															
0x1FF90																
0x1FFA0																
0x1FFB0								Page 991								
0x1FFC0								(128 bytes)								
0x1FFD0																
0x1FFE0																
0x1FFF0																E

Note:"F" indicates the beginning address of each Page, and "E" indicates the end address of each Page.

Downloaded from Arrow.com.

24.3.2 Sector Erase (partial erase in units of a Sector)

This command sequence erases the flash memory in units of a Sector. One execution of the command sequence can erase one Sector, which consists of four sectors; AREA D0, D1, C0 and C1. 28Kbytes of memory is allocated to AREA D0, 32Kbytes to AREA D1, C0 and C1.(Figure 24-1)

The Sector to be erased is specified by the 6th Bus Write Cycle address. The beginning address of the Sector to be erased is specified by the 6th address. Set the addresses specified by the instruction must be set following the Table 24-5, and also set FLSCR1<FAREA> . For example, to erase 32Kbytes from AREA C0, set "0y11" to FLSCR1<FAREA>, set "0xD5" to FLSCR2<CR1EN>, and then specify 0x08000 as the 6th Bus Write Cycle. The Sector Erase command is effective only in serial PROM mode and MCU mode, and it cannot be used in parallel PROM mode.

The time needed to erase 32K (28K) bytes of the flash memory (the time between reception of the 6th command and the moment when the value of FLSCRM<BUSY> becomes "0") is 100 ms (typ.). Other command sequences or an operation to read from the flash memory must not be executed until the erasing operation is completed. To check the completion of the erase operation, perform a read operation on FLSCRM<BUSY>, and perform polling until "0" is read from the flash memory.

Data in the erased area is 0xFF.

Note: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

Note 1: Do not perform a Sector Erase on areas other than those shown in Table 24-5.

Note 2: Even if SA is not the beginning address of the Sector, a write is performed to the Sector including the address.

Table 24-5 Addresses Range Speciable (SA)

Erase Area		FLSCR1 <FAREA>	6th address (SA)
AREA D0 (Data Aea)	0x01000 through 0x07FFF	01	0x9000
AREA D1 (Data Area)	0x08000 through 0x0FFFF	00	0x8000
AREA C0 (Code Area)	0x10000 through 0x17FFF	11	0x8000
AREA C1 (Code Area)	0x18000 through 0x1FFFF	10	0x8000

Downloaded from Arrow.com.

24.3.3 Chip Erase (all erase)

This command sequence erases the entire flash memory.

The time needed to erase all areas (the time between the reception of the 6th command and the moment when the value of FLSCRM<BUSY> becomes "0") is 400 ms (typ.). Other command sequences or an operation to read from the flash memory must not be executed until the erasing operation is completed. To check the completion of the erase operation, perform a read operation to FLSCRM<BUSY>, and perform polling until "0" is read from the flash memory.

Data in the erased area is 0xFF.

Note: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

Downloaded from Arrow.com.

24.3.4 Security Program

This command sequence sets the read protection and the write protection to the flash memory. The Security Program consists of 3-bit Security bits (SB2 through SB0), and each command sequence can set 1-bit Security bit. To set all three Security bits, command sequence must be performed three times. Once Security bits (SB2 through SB0) is set, access limitation as shown below is set to each bit.

- MCU mode

  Once SB2 is set, execution of a command sequence of the Page Program initiated by the CPU is canceled. Limitations are not imposed on read operations initiated by the CPU.

- Serial PROM mode

  Once SB2 is set, execution of a command sequence of the Page Program initiated by the CPU is canceled. Limitations are not imposed on read operations initiated by the CPU.

  During the operation through a serial communication, if a Security is set to one of the addresses from SB2 through SB0, execution of the following commands are limited: a write command to the flash memory initiated by an external device, flash memory read command and RAM loader command.

- Parallel PROM mode

  Once the setting is applied to SB0 and SB1, a command sequence of the Page Program cannot be executed in all areas of the flash memory. If a read operation is performed, 0x9898 is always read from the data bus, irrespective of the flash memory settings. If the security program setting is applied to one of SB0 or SB1, Security program is valid, but the setting must be applied to both bits.

The bits that the Security porgram is to be applied are specified in the 7th Bus Write Cycle. In this setting, the 7th address must be set according to the Table 24-6. For example, to set a Security to SB2, set 0x8100 as the 7th address.

The processing time needed to execute one command sequence (the time between the reception of the 7th command and the time when the value of FLSCRM<BUSY> becomes "0") is 1.25ms (typ.). Other command sequences or an operation to read from the flash memory must not be executed until the Security setting is completed. To check the completion of the Security setting, perform a write operation to FLSCRM<BUSY>, and then perform polling until "0" is read from the flash memory.

Note: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

Table 24-6 Access Control (SB) determined by Security Program

Security Bit	7th Address (SB)	MCU mode		Serial PROM mode		Parallel PROM mode	
		Read Protect	Write Protect	Read Protect	Write Protect	Read Protect	Write Protect
SB0	0x8000	-	-	-	-	O	O
SB1	0x8080	-	-	-	-	O	O
SB2	0x8100	-	O	-	O	-	-

Note 1: O ; Access limitation can be applied by setting the corresponding Security bit.
Note 2: - ; Security bit has no effect.

Downloaded from Arrow.com.

24.3.5 Security Erase

This command sequence disables write and read protection applied to the flash memory.

The execution of the Security Erase erases the Security bits from SB2 through SB0 at once. You cannot erase the Security bit one by one.

To check whether the security program is enabled or disabled, read 0xFF7F in product ID mode. For further details, refer to the Table 24-7. The time needed to erase it is 100ms (typ.). Other command sequences or an operation to read from the flash memory must not be performed until the settings for the Security Erase is completed. To check the completion of the erase operation, perform a read operation to FLSCRM<BUSY>, and then perform polling until "0" is read from the flash memory.

Note: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

Downloaded from Arrow.com.

24.3.6 Product ID Entry

This command activates the Product ID mode. If an instruction to read the flash memory is executed in Product ID, Security status can be read from the flash memory.

Table 24-7 Values to be read in Product ID mode

Address	Meaning	Read value	
0x0FF7F	Security status	Bit 0 (SB0)	0 : SB0 enabled 1 : SB0 disabled
		Bit 1 (SB1)	0 : SB1 enabled 1 : SB1 disabled
		Bit 2 (SB2)	0 : SB2 enabled 1 : SB2 disabled
		Bit 7 to 3	0y11111

Note:X ; Don't care

24.3.7 Product ID Exit

This command sequence is used to exit the Product ID mode.

Downloaded from Arrow.com.

24.4 Access to the Flash Memory Area

A read or a program fetch cannot be performed on the whole of the flash memory area if the following operations are performed: (a) Data is being written to the flash memory. (b) Data in the flash memory is being erased. (c) Security settings are being made in the flash memory. When performing these operation on the flash memory area, the flash memory cannot be directly accessed by using a program in the flash memory; the flash memory must be accessed using a program in the BOOTROM area or the RAM area.

Data can be written to and read from the flash memory area is in units of one Page (128Bytes). Data in the flash memory can be erased in units of 32Kbytes (partially 28Kbytes) or the whole of the flash memory at one stroke. A read can be performed using one memory transfer instruction. A write or erase, however, must be performed with the execution of the memory transfer instructions several times to several hundreds of times because the command sequence method is used. For further information about the command sequence, refer to Table 24-1.

Note 1: To allow a program to resume control on the flash memory area that is rewritten, it is recommended that you let the program jump (return) after verifying the program has been written properly.

Note 2: Do not reset the MCU (including a reset generated due to the internal factors) when data is being written to the flash memory, data is being erased from the flash memory or the security command is being executed. If a reset occurs, there is the possibility that data in the flash memory may be rewritten to an unexpected value.

Note 3: The flash memory must be accessed when the power supply of the flash memory is provided (when SDWCR1 <FLSOFF>="0" and SDWCR1<FLSWUE>="1").

Note 4: In executing erase and write operations on the Flash memory (including API operations of the BOOTROM),command sequences or verify operations cannot be performed on addresses 0x0FC00 to 0x0FFFF when SDWCR1 <DADIS> is cleared to "0". Therefore, to erase or program the Flash memory, SDWCR1<DADIS> must be set to "1" to disable the mapping of shadow RAM in the data area.

Note 5: Check the FLSCRM<BUSY> to make sure that each command sequence ends, and then hold for 200μs or more before reading data from Flash memory or starting instruction fetch.

24.4.1 Flash memory control in serial PROM mode

The serial PROM mode is used to access the flash memory by using a control program provided in the BOOTROM area. Since almost all operations relating to access to the flash memory can be controlled simply using data supplied through the serial interface (UART or SIO), it is not necessary for the user to operate the control register. For details of the serial PROM mode, see "Serial PROM mode".

To access the flash memory in serial PROM mode by using a user-specific program or peripheral functions other than UART and SIO, it is necessary to execute a control program in the RAM area by using the RAM loader command of the serial PROM mode. How to execute this control program is described in "24.4.1.1 How to transfer and write a control program to the RAM area in RAM loader mode of the serial PROM mode".

24.4.1.1 How to transfer and write a control program to the RAM area in RAM loader mode of the serial PROM mode

How to execute a control program in the RAM area in serial PROM mode is described below. A control program to be executed in the RAM area must be generated in the Intel-Hex format and be transferred using the RAM loader of the serial PROM mode.

Step 1 and 2 shown below are controlled by a program in the BOOTROM, and other steps are controlled by a program transferred to the RAM area. The following procedure is linked with a program example to be explained later.

1. Transfer the write control program to the RAM are in RAM loader mode.
2. Jump to the RAM area.
3. Set a nonmaskable interrupt vector in the RAM area.
4. Set "0y101" on FLSCR1<FLSMD>, and specify the area to be erased by making the appropriate FLSCR1<FAREA> setting. Then set "0xD5" on FLSCR2<CR1EN>.

Downloaded from Arrow.com.

5. Execute the erase command sequence.

6. Perform a read operation repeatedly until the value of FLSCRM<BUSY> becomes "0".

7. Specify the area (area erased in the step5 above) to which data is written by making the appropriate FLSCR1<FAREA> setting. Then set "0xD5" on FLSCR2<CR1EN>.

8. Execute the write command sequence.

9. Perform a read operation repeatedly until the value of FLSCRM<BUSY> becomes "0".

10. Wait time for 200µs or more.

11. Set FLSCR1<FLSMD> to “0y010”, and then set "0xD5" on FLSCR2<CR1EN> (to disable the execution of the command sequence)

Note 1: If the RAM loader is used in serial PROM mode, the BOOTROM disables (DI) a maskable interrupt, and the interrupt vector area is assigned as a RAM area (SYSCR3<RVCTR> = "1").Considering that a nonmaskable interrupt may be generated unexpectedly, it is recommended that vector address corresponding these interrupts (INTUNDEF, INTSWI : 0x001F8 to 0x001F9, WDT : 0x001FC to 0x001FD) be established and that an interrupt service routine be defined inside the RAM area.

Note 2: If a certain interrupt is used in the RAM loader program, a vector address corresponding to that interrupt and the interrupt service must be established inside the RAM area. In this case, it is recommended that a nonmaskable interrupt be handled as explained in Note 1.

Note 3: Do not set SYSCR3<RVCTR> to "0" by using the RAM loader program. If an interrupt occurs with SYSCR3<RVCTR> set to "0", the BOOTROM area is referenced as a vector address and, therefore, the program will not function properly.

(Example) A case in which a program is transferred to RAM using the RAM loader program, 128-byte data is stored in the RAM (address 0x00200 to 0x0027F) via the serial interface, the sector erase is performed on address 0x08000 through 0x0FFFF in the data area, and then write the stored RAM data to address 0x08500 through 0x0857F. If nonmaskable interrupts (INTSWI, INTUNDEF and INTWDT) occur during a write or erase operation, then the system clock reset will be generated.

```
main section code abs = 0x0100
; #### Set a nonmaskable interrupt vector inside the RAM area. #### (STEP 3)
            LD      HL,0x01FC           ; Set INTUNDEF, INTSWI interrupt vector
            LDW     (HL),sINTSWI
            LD      HL,0x01F8           ; Set INTWDT interrupt vector
            LDW     (HL),sINTWDT
; #### Prepare data to be written ####
            Store 128-byte of data in the RAM (0x0200 - 0x027F) through a data channel, such as the serial interface
; #### Disable mapping of shadow RAM in the data area ####
            LD      (SDWCR1),0x60       ; DADIS=1
; #### Sector Erase and write process ####
            LD      IY,0xF555           ; Variable for command sequence
            LD      DE,0xFAAA           ; Variable for command sequence
; Sector Erase process (STEP 5)
            LD      C,0x00              ; Set upper address
            LD      IX,0x8000           ; Set middle and lower addresses
            CALL    sSectorErase        ; Sector Erase process (0x08000)
; Write process (STEP 8)
            LD      C,0x00              ; Set upper address
            LD      IX,0x8500           ; Set middle and lower address
```


Downloaded from Arrow.com.

```
                LD      HL, 0x0200          ; Set the beginning address of RAM to be copied
                CALL    sPageProgram        ; Write process (0x08500)
; #### Enable mapping of shadow RAM in the data area ####
                LD      (SDWCR1),0x40       ; DADIS=0
; #### Execute the next main program ####
                :       :                   ; Execute the main program
                J       XXXXX
; #### Proram to be executed in RAM ####
sSectorErase:   CALL    sAddConv            ; Address conversion process
; Sector Erase process
                LD      (IY),0xAA           ; 1st Bus Write Cycle (Note1)
                LD      (DE),0x55           ; 2nd Bus Write Cycle (Note1)
                LD      (IY),0x80           ; 3rd Bus Write Cycle (Note1)
                LD      (IY),0xAA           ; 4th Bus Write Cycle (Note1)
                LD      (DE),0x55           ; 5th Bus Write Cycle (Note1)
                LD      (IX),0x30           ; 6th Bus Write Cycle (Note1)
                J       sRAMopEnd
; Write process
sPageProgram:   CALL    sAddConv            ; Convert address
                LD      (IY), 0xAA          ; 1st Bus Write Cycle (Note1)
                LD      (DE), 0x55          ; 2nd Bus Write Cycle (Note1)
                LD      (IY), 0xA0          ; 3rd Bus Write Cycle (Note1)
sPageLoop:      LD      B, (HL)
                LD      (IX),B              ; 4th through 131th Bus Write Cycle (Note1)
                INC     HL
                CMP     L, 0x80
                J       NZ, sPageLoop       ; Loop until the data of 1Page is set
; End process
sRAMopEnd       NOP
                NOP
                NOP
sLOOP1:         TEST    (FLSCRM).7          ; (STEP 6,9)
                J       F,sLOOP1            ; Loop until FLSCRM<BUSY> becomes "0".

; Wait time for 200us or more (STEP 10)
                LD      BC, 0x0216          ; @fc=16MHz (0x0216=534)
                                            ; 62.5ns x 6cycle x 534 = 200us
                                            ; To change the number by using clock
sLOOP2:         DEC     BC                  ; 2cycle
                J       F, sLOOP2           ; 4cycle

                LD      (FLSCR1),0x40       ; Disable the execution of command sequence (STEP 11)
                LD      (FLSCR2),0xD5       ; Reflect the FLSCR1 setting
                RET
; Convert address (STEP 4 and 7)
sAddConv:       LD      WA,IX
                SWAP    C
                AND     C,0x10
                SWAP    W
                AND     W,0x08
                OR      C,W
```


Downloaded from Arrow.com.

```
                XOR     C,0x08
                SHRC    C
                OR      C,0xA0
                LD      (FLSCR1),C          ; Enable the execution of command sequence. Make the FAREA setting
                LD      (FLSCR2),0xD5       ; Reflect the FLSCR1 setting
                LD      WA,IX
                TEST    C.3
                J       Z,sAddConvEnd
                OR      W,0x80
                LD      IX,WA
sAddConvEnd:    RET
; Interrupt sesrvice routine
sINTWDT:
sINTSWI:        TEST    (FLSCRM).7
                J       F,sINTWDT           ; Loop until the value of the FLSCRM<BUSY> becomes "0".
                LD      (SYSCR2),0x10       ; Generatesystem clock reset
                RETN
```

Note 1: In using a write instruction in the xxx Bus Write Cycle, make sure that you use a write instruction of more than four machine cycles or arrange write instruction in such a way that they are generated at intervals of four machine cycles or more machine cycles. If a 16-bit transfer instruction is used or if write instructions are executed at intervals of three machine cycles, the flash memory command sequence will not be transmitted properly, and a malfunction may occur.

Downloaded from Arrow.com.

24.4.2 Flash memory control in MCU mode

In MCU mode, a write operation can be performed on the flash memory by executing a control program in the RAM or the shadow RAM, or using a support program (API) provided in the BOOTROM.

24.4.2.1 How to write to the flash memory by transferring a control program to the RAM area

This section describes an example of how to execute a control program in RAM in MCU mode. A control program to be executed in the RAM must be acquired and stored in the flash memory or it must be imported from an outside source through a communication pin, etc. (The procedures described below are an example assumed that a program copy is provided in the flash memory.)

Steps 1 through 6, 13, 20 and 21 shown below concern the control by a program in the flash memory, and other steps concern the control by a program transferred to RAM. The following procedure is linked with a program example to be described later.

1. Set the interrupt master enable flag to "disable (DI)"(IMF ← "0").
2. Transfer the write control program to the RAM.
3. Establish the nonmaskable interrupt vector in the RAM area.
4. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "1", set "0xD4" on SYSCR4. Then allocate the RAM to the code area, and switch the vector area to the RAM area.
5. Disable mapping of shadow RAM in the data area.
6. Invoke the erase processing program in the RAM area by executing a CALL instruction.
7. Set FLSCR1<FLSMD> to "0y101", and specify the area to be erased by making the appropriate FLSCR1<FAREA> setting. Then set "0xD5" on FLSCR2<CR1EN>.
8. Execute the erase command sequence.
9. Perform a read repeatedly until the value of FLSCRM<BUSY> becomes "0".
10. Wait time for 200μs or more.
11. After setting FLSCR1<FLSMD> to "0y010" and FLSCR1<FAREA> to "0y00", set "0xD5" on FLSCR2<CR1EN>. (This disables the execution of the command sequence and returns FAREA to the initial state of mapping.)
12. Execute the RET instruction to return to the flash memory.
13. Invoke the write program in the RAM area by executing a CALL instruction.
14. Set FLSCR1<FLSMD> to "0y101", and make the appropriate FLSCR1<FAREA> setting to specify the area (area erased by performing step 7 above) on which a write is to be performed. Then set "0xD5" on FLSCR2<CR1EN>.
15. Execute the write command sequence.
16. Perform a read repeatedly until the value of FLSCRM<BUSY> becomes "0".
17. Wait time for 200μs or more.

Downloaded from Arrow.com.

18. After setting FLSCR1<FLSMD> to "0y010" and FLSCR1<FAREA> to "0y00", set "Dx05" on FLSCR2<CR1EN> (This disables the execution of the command sequence and returns FAREA to the initial state of mapping.)

19. Execute the RET instruction to return to the flash memory.

20. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "0", set "0xD4" on SYSCR4. Then release RAM allocation for the code area, and switch the vector area to the flash area.

21. Enable mapping of shadow RAM in the data area.

Note 1: Before writing data to the flash memory from the RAM area in MCU mode, the vector area must be switched to the RAM area by using SYSCR3<RVCTR>, data must be written to the vector addresses (INTUNDEF, INTSWI : 0x001F8 to 0x001F9, INTWDT : 0x001FC to 0x001FD) that correspond to nonmaskable interrupts, and the interrupt subroutine (RAM area) must be defined. This allows you to trap the errors that may occur due to an unexpected nonmaskable interrupt during a write operation. If SYSCR3<RVCTR> is set in the flash memory area and if an unexpected interrupt occurs during a write operation, a malfunction may occur because the vector are in the flash memory cannot be read properly.

Note 2: Before using a certain interrupt in MCU mode, the vector address corresponding to the interrupt and the interrupt service routine must be established in the RAM area. In this case, the nonmaskable interrupt setting must be made, as established in Note1.

Note 3: Before jumping from the flash memory to the RAM area, RAM must bealocated to the code area by making the appropriate SYSCR3<RAREA> setting (setting made in step 4 in the procedure described on the previous page).

Note 4: The setting described in step 4 cannot be allocated to the code area from 0x0040 to the end of RAM address in the flash memory. If the setting is executed in the code area in the flash memory, the fetch area is switched from the flash memory area to the RAM area, and the program will not function properly.

(Example) Case in which a program is transferred to RAM (starting address 0x00200), a sector erase is performed on the address 0x08000 through 0x0FFFF in the data area, and then RAM data (0x00100 through 0x0017F) is written to the address 0x08500 through 0x0857F. If nonmaskable interrupts (INTSWI, INTUNDEF or INTWDT) occur during a write or erase operation is being executed, system clock reset is generated.

```
                cRAMStartAdd equ 0x0200                 ; RAM start address
main section code abs = 0x1F000
                DI                                      ; Disable interrupts (STEP 1)
; #### Transfer the program to RAM #### (STEP2)
                LD      HL,cRAMStartAdd
                LD      IX,sRAMprogStart
sRAMLOOP:       LD      A,(IX)                          ; Transfer to cRAMStartAdd the program (from
                LD      (HL),A                          ; sRAMprogStart to sRAMprogEnd).
                INC     HL
                INC     IX
                CMP     IX,sRAMprogEnd
                J       NZ,code_addr(sRAMLOOP)
; #### Set a nonmaskable interrupt vector in the RAM area #### (STEP 3)
                LD      HL,0x01FC                       ; Set INTUNDEF and INTSWI interrupt vector
                LDW     (HL),sINTSWI - sRAMprogStart + cRAMStartAdd
                LD      HL,0x01F8                       ; Set INTWDT interrupt vector
                LDW     (HL),sINTWDT - sRAMprogStart + cRAMStartAdd
; #### Allocate RAM to the code area. Switch the vector area to RAM #### (STEP 4)
                LD      (SYSCR3),0x06                   ; Set RAREA = 1 and RVCTR = 1.
                LD      (SYSCR4),0xD4                   ; Enable Code
; #### Disable mapping of shadow RAM in the data area ####
```


Downloaded from Arrow.com.

```
                LD          (SDWCR1),0x60               ; DADIS=1 (STEP 5)
; #### Sector Erase and write process ####
                LD          IY,0xF555                   ; Variable for command sequence
                LD          DE,0xFAAA                   ; Variable for command sequence
; Sector Erase process (STEP 6)
                LD          C,0x00                      ; Set upper addresses
                LD          IX,0x8000                   ; Set middle and lower addresses
                CALL        sSectorErase - sRAMprogStart + cRAMStartAdd
                                                        ; Sector Erase process (0x1E000)
; Write process (STEP 13)
                LD          C,0x00                      ; Set upper addresses
                LD          IX,0x8500                   ; Set middle and lower addresses
                LD          HL, 0x0100                  ; Set the beginning address of RAM to be copied.
                CALL        sPageProgram - sRAMprogStart + cRAMStartAdd
                                                        ; Write process (0x1E500)
; ####Allocate RAM to the code area. Switch the vector area to RAM.#### (STEP 20)
                LD          (SYSCR3),0x00               ; Set RAREA = 0 and RVCTR = 0
                LD          (SYSCR4),0xD4               ; Enable Code
; #### Enable mapping of shadow RAM in the data area ####
                LD          (SDWCR1),0x40               ; DADIS=0 (STEP 21)
; #### Main program process ####
                :           :                           ; Main program process
                J           XXXXX
ramexe section code abs = 0x2000
; #### Program to be executed in RAM ####
sRAMprogStart:
                NOP                                     ; Fail-safe process
                NOP
                NOP
                NOP
                NOP
                LD          (SYSCR2),0x10               ; Generate system clock reset
sSectorErase:   CALL        sAddConv - sRAMprogStart + cRAMStartAdd
                                                        ; Address conversion process (STEP 7)
; Sector Eraseprocess (STEP 8)
                LD          (IY),0xAA                   ; 1st Bus Write Cycle (Note1)
                LD          (DE),0x55                   ; 2nd Bus Write Cycle (Note1)
                LD          (IY),0x80                   ; 3rd Bus Write Cycle (Note1)
                LD          (IY),0xAA                   ; 4th Bus Write Cycle (Note1)
                LD          (DE),0x55                   ; 5th Bus Write Cycle (Note1)
                LD          (IX),0x30                   ; 6th Bus Write Cycle (Note1)
                J           sRAMopEnd
; Write process (STEP 13)
sPageProgram:   CALL        sAddConv - sRAMprogStart + cRAMStartAdd
                                                        ; Address conversion process (STEP 14)
                LD          (IY), 0xAA                  ; 1st Bus Write Cycle (Note1)
                LD          (DE), 0x55                  ; 2nd Bus Write Cycle (Note1)
                LD          (IY), 0xA0                  ; 3rd Bus Write Cycle (Note1)
sPageLoop:      LD          B, (HL)
                LD          (IX),B                      ; 4th through 131th Bus Write Cycle (Note1)
                INC         HL
```


Downloaded from Arrow.com.

```
                CMP     L, 0x80
                J       NZ, sPageLoop           ; Loop until data of one Page is set
; End process
sRAMopEnd:      NOP                             ;
                NOP                             ;
                NOP                             ;
sLOOP1:         TEST    (FLSCRM).7              ; (STEP 9 and 16)
                J       F,sLOOP1                ; Loop until FLSCRM<BUSY>becomes "0".

; Wait time for 200us or more (STEP 10, 17)
                LD      BC, 0x0216              ; @fc=16MHz (0x0216=534)
                                                ; 62.5ns x 6cycle x 534 = 200us
                                                ; To change the number by using clock
sLOOP2:         DEC     BC                      ; 2cycle
                J       F,sLOOP2                ; 4/2cycle

                LD      (FLSCR1),0x40           ; Disable the execution of command sequence
                                                ; (STEP 11, 18)
                LD      (FLSCR2),0xD5           ; Reflect the FLSCR1 setting
                RET                             ; Return to flash memory
; Address conversion process (STEP 7, 14)
sAddConv:       LD      WA,IX
                SWAP    C
                AND     C,0x10
                SWAP    W
                AND     W,0x08
                OR      C,W
                XOR     C,0x08
                SHRC    C
                OR      C,0xA0
                LD      (FLSCR1),C              ; Enable the execution of command sequence.Make the FAREA setting
                LD      (FLSCR2),0xD5           ; Reflect the FLSCR1 setting
                LD      WA,IX
                TEST    C.3
                J       Z,sAddConvEnd
                OR      W,0x80
                LD      IX,WA
sAddConvEnd:    RET
; Interrupt service routine
sINTWDT:
sINTSWI:        TEST    (FLSCRM).7
                J       F,sINTWDT               ; Loop until FLSCRM<BUSY> becomes "0".
                LD      (SYSCR2),0x10           ; Generate system clock reset
                RETN
sRAMprogEnd:    NOP
```

Note 1: In using a write instruction in the xxx Bus Write Cycle, make sure that you use a write instruction of more than four machine cycles or arrange write instructions in such a way that they are generated at intervals of four or more machine cycles. If a 16-bit transfer instruction is used or if write instructions are executed at intervals of two machine cycles, the flash memory command sequence will not be transmitted properly, and a malfunction may occur.

Downloaded from Arrow.com.

24.4.2.2 How to rewrite the flash memory by using a support program (API) of BOOTROM

This section describes how to perform a rewrite on the flash memory by using a support program (API) of BOOTROM in MCU mode. For details, refer to"24.5 API (Application Programming Interface)".

Steps 1 through 7 and 19 through 21 shown below concern the control by a program in the flash memory, and other steps concern the control by a program transferred to RAM. The following procedure is linked with a program example to be described later.

1. Set the interrupt master enable flag to "disable (DI)"(IMF ← "0").
2. Transfer the write control program to the RAM.
3. Allocate BOOTROM to the data/code area. (FLSCR1<BAREA>="1")
4. Establish the nonmaskable interrupt vector in the .BTReset (0x1000).
5. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "1", set "0xD4" on SYSCR4. then allocate RAM to the code area, and switch the vector area to the RAM area.
6. Disable mapping of shadow RAM in the data area. (SDWCR1<DADIS>="1")
7. Invoke the erase and write processing program in the RAM area by executing a CALL instruction.
8. Set the address range to be erased to the A register according to Table 24-9.
9. Set "0xD5" to the C register as Enable Code.
10. CALL .BTEraseSec(0x1012).(Sector Erase is performed.
11. Enable the execution of command sequence.
12. Set "0xD5" on FLSCR2 after setting "1" on FLSCR1<BAREA>.
13. Execute the erase command sequence.
14. Perform a read repeatedly until the value of FLSCRM<BUSY> becomes "0".
15. Wait time for 200μs or more.
16. Disable the erase command sequence.
17. Returns FAREA to the initial state of mapping.
18. Execute the RET instruction to return to the flash memory.
19. Set "0xD5" to FLSCR2 after setting "0" to FLSCR1<BAREA>. (Hide BOOTROM)
20. Set "0xD4" to SYSCR4 adter setting "0" to both SYSCR3<RAREA> and SYSCR3<RVCTR>.
21. Enable mapping of shadow RAM in the data area.

Downloaded from Arrow.com.

(Example) Case in which a program is transferred to RAM (starting address 0x00200), a sector erase is performed on the address 0x08000 through 0x0FFFF in the data area, and then RAM data (0x00100 through 0x0017F) is written to the address 0x08000 through 0x0857F. If nonmaskable interrupts (INTSWI, INTUNDEF or INTWDT) occur during a write or erase operation is going on, system clock reset is generated.

```
                cRAMStartAdd    equ 0x0200          ; RAM Start address

                .BTReset        equ 0x1000          ; Generate system clock reset
                .BTEraseSec     equ 0x1012          ; Sector Erase
                .BTEraseChip    equ 0x1014          ; Chip Erase
                .BTGetSP        equ 0x1016          ; Check the status of Security Program
                .BTSetSP        equ 0x1018          ; Configure the Security Program
                .BTRead         equ 0x101A          ; Read data from the flash memory
                .BTConvAdr      equ 0x101C          ; Address conversion process
                .BTCalcUART     equ 0x101E          ; Calculate the setting value of UART from the timercapture value
                .BTErsSP        equ 0x1020          ; Disable Security Program
                .BTUpdSD        equ 0x1022          ; Shadow RAM update

main section code abs = 0x1F000
; #### Disable interupts ####                       (STEP 1)
                DI                                  ; IMF="0"

; #### Transfer the program to RAM ####             (STEP 2)
                LD      HL,cRAMStartAdd             ; RAM start address
                LD      IX,sRAMprogStart            ;
sRAMLOOP:       LD      A,(IX)                      ; Transfer to cRAMStartAdd the program
                LD      (HL),A                      ; (from sRAMprogStart to sRAMprogEnd)
                INC     HL
                INC     IX
                CMP     IX,sRAMprogEnd
                J       NZ,code_addr(sRAMLOOP)
; #### Allocate BOOTROM to the data/code area #### (STEP 3)
                LD      (FLSCR1),0x50               ; Set BAREA to "1"
                LD      (FLSCR2),0xD5               ; Reflect the FLSCR1 setting
; #### Set a nonmaskable interrupt vector to .BTReset #### (STEP 4)
                LD      HL,0x01FC                   ; Set INTUNDEF and INTSWI interrupt vectors
                LDW     (HL),.BTReset
                LD      HL,0x01F8                   ; Set INTWDT interrupt vector
                LDW     (HL),.BTReset
; #### Allocate RAM in the code area. Switch the vector area to RAM #### (STEP 5)
                LD      (SYSCR3),0x06               ; Set RAREA = 1 and RVCTR = 1
                LD      (SYSCR4),0xD4               ; Enable Code
; #### Disable mapping of shadow RAM in the data area #### (STEP 6)
                LD      (SDWCR1),0x60               ; DADIS=1

; #### Sector Erase / Write process #### (STEP 7)
                CALL sSectorErase - sRAMprogStart + cRAMStartAdd ;

; #### Hide BOOT ROM #### (STEP 19)
                LD      (FLSCR1), 0x40              ; BAREA = "0"
```


Downloaded from Arrow.com.

```
                LD          (FLSCR2), 0xD5          ; Enable Code

; #### Allocate RAM to the code area. Switch the vector area to RAM #### (STEP 20)
                LD          (SYSCR3),0x00           ; Set RAREA = 0 and RVCTR = 0
                LD          (SYSCR4),0xD4           ; Enable Code

; #### Enable mapping of shadow RAM in the data area #### (STEP 21)
                LD          (SDWCR1),0x40           ; DADIS=0
; #### Main program process ####
                :           :                       ; Main program process
                J           XXXX


; ------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
; Below is the program transferred to the RAM.
; ------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
ramexe section code abs = 0x2000

; #### RAM program ####
sRAMprogStart:
                NOP                                 ; Fail-safe process
                NOP
                NOP
                NOP
                NOP
                LD          (SYSCR2),0x10           ; Generate system clock reset

; #### Sector Erase process (API) ####
sSectorErase:
; #### Set erase area ####                         (STEP 8, 9)
                LD          A, 0x20                 ; Specify the area to be erased (0x08000 through 0x0FFFF)
                LD          C, 0xD5                 ; Enable Code
; #### Execute Sector Erase ####                   (STEP 10)
                CALL        (.BTEraseSec)           ; Execute Sector Erase
; #### Write process ####
sPageProgram:
; #### Write process ####                          (STEP 11, 12)
                LD          IY,0xF555               ; Variable for command sequence
                LD          DE,0xFAAA               ; Variable for command sequence

                LD          C, 0x00                 ; Set upper address
                LD          IX,0x8000               ; Set middle and low address
                LD          HL, 0x0100              ; Set the biginning address of RAM to be copied.
                CALL sAddConv-sRAMprogStart+cRAMStartAdd ; Address conversion process

; #### Write process ####                          (STEP 13)
                LD          (IY), 0xAA              ; 1st Bus Write Cycle (Note1)
                LD          (DE), 0x55              ; 2nd Bus Write Cycle (Note1)
                LD          (IY), 0xA0              ; 3rd Bus Write Cycle (Note1)
sPageLoop:      LD          B, (HL)
                LD          (IX),B                  ; 4th ~ 131th Bus Write Cycle (Note1)
```


Downloaded from Arrow.com.

```
                INC     HL
                CMP     L, 0x80
                J       NZ, sPageLoop           ; Loop untill data of one page is set

; End process
sRAMopEnd:      NOP                             ;
                NOP                             ;
                NOP                             ;
sLOOP1:         TEST    (FLSCRM).7              ; (STEP 14)
                J       F,sLOOP1                ; Loop untill FLSCRM<BUSY> becomes "0"

                ; Wait time for 200us or more (STEP 15)
                LD      BC, 0x0216              ; @fc=16MHz (0x216=534)
                                                ; 62.5ns x 6cycle x 534 = 200μs
                                                ; To change the number by using clock
sLOOP2:         DEC     BC                      ; 2cycle
                J       F, sLOOP2               ; 4/2cycle

                LD      (FLSCR1),0x40           ; Disable the execution of command sequence (STEP 16, 17)
                LD      (FLSCR2),0xD5           ; Enable code

                RET                             ; Return to the flash memory (STEP 18)


; Address conversion process (STEP 11, 12)
sAddConv:       LD      WA,IX
                SWAP    C
                AND     C,0x10
                SWAP    W
                AND     W,0x08
                OR      C,W
                XOR     C,0x08
                SHRC    C
                OR      C,0xA0
                LD      (FLSCR1),C              ; Enable the execution of command sequence. Make the FAREA setting
                LD      (FLSCR2),0xD5           ; Enable code
                LD      WA,IX
                TEST    C.3
                J       Z,sAddConvEnd
                OR      W,0x80
                LD      IX,WA
sAddConvEnd:    RET

sRAMprogEnd:    NOP
```

Note 1: In using a write instruction in the xxx Bus Write Cycle, make sure that you use a write instruction of more than four machine cycles or arrange write instructions in such a way that they are generated at intervals of four or more machine cycles. If a 16-bit transfer instruction is used or if write instructions are executed at intervals of two machine cycles, the flash memory command sequence will not be transmitted properly, and a malfunction may occur.

Downloaded from Arrow.com.

Note 2: Do not allocate the above program to the address from 0x0000 through 0x17FF in the code area of the flash memory. If the area is set as BAREA="1", it changes from the flash memory area to the BOOTROM area, so that the program will not function properly and the microcontroller may malfunction.

Note 3: It is not necessary to add DI instruction to disable interrupt maskable enable flag (IMF) for above sample program, because the support program has it. However, the support program does not include EI instruction. Enable IMF after finishing all above process, if you use interrupt process.

Downloaded from Arrow.com.

24.4.2.3 How to set the security program by using a support program (API) of BOOTROM

This section describes how to perform a security program on the flash memory by using a support program (API) of BOOTROM in MCU mode. For details, refer to"24.5 API (Application Programming Interface)".

Steps 1 through 7 and 14 through 16 shown below concern the control by a program in the flash memory, and other steps concern the control by a program transferred to RAM. The following procedure is linked with a program example to be described later.

1. Set the interrupt master enable flag to "disable (DI)"(IMF ← "0").
2. Transfer the write control program to the RAM.
3. Allocate BOOTROM to the data/code area. (FLSCR1<BAREA>="1")
4. Establish the nonmaskable interrupt vector in the .BTReset (0x1000).
5. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "1", set "0xD4" on SYSCR4. then allocate RAM to the code area, and switch the vector area to the RAM area.
6. Disable mapping of shadow RAM in the data area. (SDWCR1<DADIS>="1")
7. Invoke the security program in the RAM area by executing a CALL instruction.
8. Call .BTGetSP(0x1016). (After processing, security program state returns to the A register)
9. Read the A register, and then jump to the sSKIP because the security program is already set.
10. Set "0xD5" to the A register as Enable Code.
11. Call .BTSetSP(0x1018). (Security Program is performed.)
12. Wait time for 200µs or more.
13. Execute the RET instruction to return to the flash memory.
14. Set "0" on FLSCR1<BAREA>, and set "0xD5" on FLSCR2. (Hide BOOTROM)
15. After setting "0" to both SYSCR3<RAREA> and SYSCR3<RVCTR>, set "0xD4" on SYSCR4.
16. Enable mapping of shadow RAM in the data area.

(Example) Check the setting of Security Program(SB2). If it is not set yet, set Security Program(SB2).

```
cRAMStartAdd   equ 0x0200        ; RAM start address

.BTReset       equ 0x1000        ; Generate system clock reset.
.BTEraseSec    equ 0x1012        ; Sector Erase
.BTEraseChip   equ 0x1014        ; Chip Erase
.BTGetSP       equ 0x1016        ; Check the state of Security Program
.BTSetSP       equ 0x1018        ; Set Security Program
.BTRead        equ 0x101A        ; Read data from the flash memory
.BTConvAdr     equ 0x101C        ; Address conversion
.BTCalcUART    equ 0x101E        ; Calculate the setting value of UART from the timercapture value
```


Downloaded from Arrow.com.

```
                .BTErsSP    equ 0x1020          ; Disable Security Program
                .BTUpdSD    equ 0x1022          ; Shadow RAM update

main section code abs = 0x1F000
; #### Disable interupts ####                   (STEP 1)
                DI                              ; IMF="0"

; #### Transfer the program to RAM ####         (STEP 2)
                LD          HL,cRAMStartAdd     ; RAM start address
                LD          IX,sRAMprogStart    ;
sRAMLOOP:       LD          A,(IX)              ; Transfer to cRAMStartAdd the program
                LD          (HL),A              ; (from sRAMprogStart to sRAMprogEnd)
                INC         HL
                INC         IX
                CMP         IX,sRAMprogEnd
                J           NZ,code_addr(sRAMLOOP)

; #### Allocate BOOTROM to the data/code area ####   (STEP 3)
                LD          (FLSCR1),0x50       ; Set BAREA to "1"
                LD          (FLSCR2),0xD5       ;Enable Code

; #### Set a nonmaskable interrupt vector to .BTReset ####   (STEP 4)
                LD          HL,0x01FC           ; Set INTUNDEF and INTSWI interrupt vectors
                LDW         (HL),.BTReset
                LD          HL,0x01F8           ; Set INTWDT interrupt vector
                LDW         (HL),.BTReset

; #### Allocate RAM in the code area. Switch the vector area to RAM #### (STEP5)
                LD          (SYSCR3),0x06       ; Set RAREA = 1 and RVCTR = 1
                LD          (SYSCR4),0xD4       ; Enable Code

; #### Disable mapping of shadow RAM in the data area ####   (STEP 6)
                LD          (SDWCR1),0x60       ; DADIS=1

; #### Security program process ####            (STEP 7)
                CALL sSecProg - sRAMprogStart + cRAMStartAdd ;


; #### Hide BOOT ROM #### (STEP 14)
                LD          (FLSCR1), 0x40      ; BAREA = "0"
                LD          (FLSCR2), 0xD5      ; Enable Code

; #### Allocate RAM to the code area. Switch the vector area to RAM #### (STEP 15)
                LD          (SYSCR3),0x00       ; Set RAREA = 0 and RVCTR = 0
                LD          (SYSCR4),0xD4       ; Enable Code

; #### Enable mapping of shadow RAM in the data area ####   (STEP 16)
                LD          (SDWCR1),0x40       ; DADIS=0

; #### Main program process ####
                :           :                   ; Main program process
```


Downloaded from Arrow.com.

```
                J           XXXX

; ------------------------------------------------------------------------------------------------------------------------------------------------------
;Below is the program transferred to the RAM
; ------------------------------------------------------------------------------------------------------------------------------------------------------
ramexe section code abs = 0x2000

; #### RAM program ####
sRAMprogStart:
                NOP                                 ; Fail-safe process
                NOP
                NOP
                NOP
                NOP
                LD          (SYSCR2),0x10           ; Generate system clock reset

; #### Security Program process (API) ####
sSecProg:
; #### Check the status of security program ####  (STEP 8)
                CALL        (.BTGetSP)              ; Check the status of security program
                TEST        A.2                     ; (STEP 9)
                J           T,sSKIP                 ; Go to sSKIP if SB2 is "0"
; #### Security Program enable process (API) ####
                LD          WA,0x8000
                LD          DE,0x0080
                LD          (SP-),0xD5              ; Enable Code (STEP 10)
                LD          C,0xFB                  ; Set SB2=0
                CALL        (.BTSetSP)              ; Set Security Program

; Wait time for 200us or more (STEP 12)
                LD          BC, 0x0216              ; @fc=16MHz (0x216=534)
                                                    ; 62.5ns x 6cycle x 534 = 200μs
                                                    ; To change the number by using clock
sLOOP2:         DEC         BC                      ; 2cycle
                J           F, sLOOP2               ; 4/2cycle

sSKIP:          LD          (FLSCR1),0x40           ; Set BAREA to "0"
                LD          (FLSCR2),0xD5           ; Enable Code
                LD          (SYSCR3),0x00           ; Set RAREA = 0 and RVCTR = 0
                LD          (SYSCR4),0xD4           ; Enable Code

                RET                                 ; Return to the flash memory (STEP 13)

sRAMprogEnd:    NOP
```

Note 1: Do not allocate the above program to the address from 0x0000 through 0x17FF in the code area of the flash memory. If the area is set as BAREA="1", it changes from the flash memory area to the BOOTROM area, so that the program will not function properly and the microcontroller may malfunction.

Note 2: It is not necessary to add DI instruction to disable interrupt maskable enable flag (IMF) for above sample program, because the support program has it. However, the support program does not include EI instruction. Enable IMF after finishing all above process, if you use interrupt process.

Downloaded from Arrow.com.

24.4.2.4 How to rewrite the program itself by using the shadow RAM and the support program (API) of the BOOTROM

The following shows an example of how to execute a control program stored in the shadow RAM in MCU mode.

Step 1 through 18 shown below are the processes performed by the program stored in the shadow RAM.

1. Set the interrupt master enable flag to "disable (DI)"(IMF ← "0").
2. Allocate BOOTROM to the data/code area. (FLSCR1<BAREA>="1")
3. Establish the nonmaskable interrupt vector in the .BTReset (0x1000).
4. After setting both SYSCR3<RAREA> and SYSCR3<RVCTR> to "1", set "0xD4" on SYSCR4. then allocate RAM to the code area, and switch the vector area to the RAM area.
5. Disable mapping of shadow RAM in the data area. (SDWCR1<DADIS>="1")
6. Set the range of address to be erased into the A register according to Table 24-9.
7. Set "0xD5" to the C register as Enable Code.
8. Call .BTEraseSec (0x1012). (Sector Erase is performed. It is not necessary from the step 4 to the step6, if programming area is already erased.)
9. Store a new program in RAM through a data channel, such as the serial interface.
10. Enable the execution of command sequence.
11. Set "0xD5" on FLSCR2 after setting "1" on FLSCR1<BAREA>.
12. Execute the write command sequence.
13. Perform a write repeatedly until the value of FLSCRM<BUSY> becomes "0".
14. Wait time for 200µs or more.
15. Disable the write command sequence.
16. Returns FAREA to the initial state of mapping.
17. Set the address to return after updating the shadow RAM into the WA register.
18. Call .BTUpdSD(0x1022). (Shadow RAM is updated.)

(Example) Case in which a Sector Erase is performed to the address from 0x18000 through 0x1FFFF in the shadow RAM area, and then write the new program obtained via the serial interface to the address 0x18000 through 0x1FFFF in the flash memory. Then, update the shadow RAM, and run the new program from the address 0x1F900. If nonmaskable interrupts (INTSWI, INTUNDEF or INTWDT) occur during a write or erase operation is going on, system clock reset is generated.

```
cRAMStartAdd   equ 0x0200          ; RAM Start address

.BTReset       equ 0x1000          ; Generate system clock reset
```


2026-04-21
Downloaded from Arrow.com.

```
                .BTEraseSec     equ 0x1012          ; Sector Erase
                .BTEraseChip    equ 0x1014          ; Chip Erase
                .BTGetSP        equ 0x1016          ; Check the status of Security Program
                .BTSetSP        equ 0x1018          ; Configure the Security Program
                .BTRead         equ 0x101A          ; Read data from the flash memory
                .BTConvAdr      equ 0x101C          ; Address conversion process
                .BTCalcUART     equ 0x101E          ; Calculate the setting value of UART from the timercapture value
                .BTErsSP        equ 0x1020          ; Disable Security Program
                .BTUpdSD        equ 0x1022          ; Shadow RAM update

ram section data abs = 0x00200
                .gPrgAdd dsb 2                      ; Variable for write address

main section code abs = 0x1F000
; #### Disable interupts ####                       (STEP 1)
                DI                                  ; IMF="0"

; #### Allocate BOOTROM to data/code area ####      (STEP 2)
                LD      (FLSCR1),0x50               ; Set "1" to BAREA
                LD      (FLSCR2),0xD5               ; Reflict the FLSCR1 setting

; #### Set a nonmaskable interrupt vector to .BTReset ####   (STEP 3)
                LD      HL,0x01FC                   ; Set INTUNDEF and INTSWI interrupt vector
                LDW     (HL),.BTReset
                LD      HL,0x01F8                   ; Set INTWDT interrupt vector
                LDW     (HL),.BTReset

; #### Allocate RAM to the code area. Switch the vector area to RAM #### (STEP4)
                LD      (SYSCR3),0x06               ; Set RAREA = 1 and RVCTR = 1
                LD      (SYSCR4),0xD4               ; Enable Code

; #### Disable mapping of shadow RAM in the data area ####   (STEP 5)
                LD      (SDWCR1),0x60               ; DADIS=1

; #### Sector Erase process ####                    (STEP 6)
; #### Set erase area ####                          (STEP 6, 7)
                LD      A, 0x20                     ; Set the address range to be erased (0x08000~0x0FFFF)
                LD      C, 0xD5                     ; Enable Code
; #### Execute Sector Erase ####                    (STEP 8)
                CALL    (.BTEraseSec)               ; Perform Sector Erase
; #### Prepare data to be program ####
                LD      HL, .gPrgAdd                ; Set variable for write address
                LDW     (HL), 0x8000                ; Set the start address
sGetPrgDat:
; Store a new program in RAM (0x00100 through 0x0017F) 128bytes at a time through a data channel,
; such as the serial interface (STEP 9)

sPageProgram:
; #### write process ####                           (STEP 10,11)
                LD      IY,0xF555                   ; Variable for command sequence
```


Downloaded from Arrow.com.

```
                LD      DE,0xFAAA       ; Variable for command sequence
                LD      IX,0x8000       ; Set page addresses
                LD      HL, (.gPrgAdd)  ; Set start address
                LD      (FLSCR1),0xB0   ; Enable the execution of command sequence. Make the
                                        FAREA setting
                LD      (FLSCR2),0xD5   ; Enable code

; #### Write sequence ####              (STEP 12)
                LD      (IY), 0xAA      ; 1st Bus Write Cycle (Note1)
                LD      (DE), 0x55      ; 2nd Bus Write Cycle (Note1)
                LD      (IY), 0xA0      ; 3rd Bus Write Cycle (Note1)
sPageLoop:      LD      B, (HL)
                LD      (IX),B          ; 4th ~ 131th Bus Write Cycle (Note1)
                INC     HL
                CMP     L, 0x80
                J       NZ, sPageLoop   ; Loop until the data of 1Page is set

; End process
sRAMopEnd:      NOP                     ;
                NOP                     ;
                NOP                     ;
sLOOP1:         TEST    (FLSCRM).7      ; (STEP 13)
                J       F,sLOOP1        ; Loop until FLSCRM<BUSY> becomes "0".

; Wait time for 200us or more (STEP 14)
                LD      BC, 0x0216      ; @fc=16MHz (0x0216=534)
                                        ; 62.5ns x 6cycle x 534 = 200us
                                        ; To change the number by using clock
sLOOP2:         DEC     BC              ; 2cycle
                J       F, sLOOP2       ; 4cycle

                LD      (FLSCR1),0x40   ; Disable the execution of command sequence (STEP 15,16)
                LD      (FLSCR2),0xD5   ; Reflect the FLSCR1 setting

; #### Shadow RAM update #### (STEP 17, 18)
                LD      WA, 0xF900      ; Set 0x1F900 as return address
                CALL    (.BTUpdSD)      ; Jump to 0x1F900 after shadow RAM update
```

Note 1: In using a write instruction in the xxx Bus Write Cycle, make sure that you use a write instruction of more than four machine cycles or arrange write instructions in such a way that they are generated at intervals of four or more machine cycles. If a 16-bit transfer instruction is used or if write instructions are executed at intervals of two machine cycles, the flash memory command sequence will not be transmitted properly, and a malfunction may occur.

Note 2: Do not allocate the above program to the address from 0x0000 through 0x17FF in the code area of the flash memory. If the area is set as BAREA="1", it changes from the flash memory area to the BOOTROM area, so that the program will not function properly and the microcontroller may malfunction.

Note 3: It is not necessary to add DI instruction to disable interrupt maskable enable flag (IMF) for above sample program, because the support program has it. However, the support program does not include EI instruction. Enable IMF after finishing all above process, if you use interrupt process.

Downloaded from Arrow.com.

24.4.2.5 How to read data from flash memory

To read data from the flash memory, execute transfer instruction (Read) for the memory. It is possible to read arbitrary data including the code area by setting the FLSCR1<FAREA> and FLSCR2.

(Example) Case in which data is read from 0x1F000 in the code area and stored at (0x98) in RAM

```
LD      (FLSCR1),0xA8      ; Select AREA C1
LD      (FLSCR2),0xD5      ; Reflect the FLSCR1 setting
LD      A,(0xF000)         ; Read data from 0xF000
LD      (0x98),A           ; Store data at 0x98
LD      (FLSCR1),0x40      ; Select AREA D0
LD      (FLSCR2),0xD5      ; Reflect the FLSCR1 setting
```


Downloaded from Arrow.com.

24.5 API (Application Programming Interface)

The BOOTROM has a support program (API) which contains a special subroutine for erasing or writing on the flash memory. After mapping of the BOOTROM, it allows easy erasing or writing on the flash memory by calling the subroutine in BOOTROM. The Table 24-8 shows the list of API.

Note 1: API must be called only when, except (.BTCalcUART), the flash memory is turned ON (when SDWCR1<FLSOFF> ="0" and SDWCR1<FLSWUE>="1"). If the flash memory is turned OFF, an adequate response cannot be obtained even if API is called.

Note 2: When sector erasing, chip erasing, security setting or security releasing is performed by API on the BOOT ROM, hold wait time for 200Éþs or more before reading data from Flash memory or starting instruction fetch after the completion of calling API.

Note 3: Calling API and holding wait time must be performed on RAM or Shadow RAM.

Note 4: When the on-chip debug emulator is used, do not stop program in the API of BOOTROM. When execution of a program is stopped in the API part of BOOTROM, the API command does not function.

Applicable functions

- The execution in API by the STEP function
- The stop in API by the Brake function

Table 24-8 List of API

Address	Contents	Using Stack (Note2)	Work register (Note1)	Argument		Return value		
				Register	Setting value	Register	Contents	
0x1000 (.BTReset)	Perform polling until FLSCRM<BUSY> becomes "0", and then generate system clock reset	-	-	-	-	-	-	-
0x1012 (.BTEraseSec)	Erasing the specified one sector.	6bytes	WA BC DE IX	A	Erase Sector	A	0xFF	Normal end
				C	0xD5 (Enable Code)		0x01	NO BUSY error
0x1014 (.BTEraseChip)	Executing the Chip Erase	8bytes	WA BC DE IX	A	0xD5 (Enable Code)	A	0xFF	Normal end
							0x01	NO BUSY error
0x1016 (.BTGetSP)	Getting the status of Security Program	8bytes	WA BC DE IX			A	the contents of ID (0x0FF7F)	
0x1018 (.BTSetSP)	Setting the Security Program	13 bytes	WA BC DE IX	WA	0x8000	A	0xFF	Normal end
				C	Security setting value		0x01	NO BUSY error
				DE	0x0080	-	0x02	Address overflow error
				(SP-)	0xD5 (Enable Code)	-	-	-
0x101C (.BTConvAdr)	Converting the flash memory address	4bytes	WA BC DE	WA	Beginning address of Page (M, L)	WA	Converted address	
				C	Beginning address of Page (H)	-	-	
0x101E (.BTCalcUART)	Calculating the UART register setting (Baud rate) using the value of the pulse width captured by timer counter	4bytes	WA BC DE IX IY	WA	Captured value by timer counter	W	Setting value for RTSEL	
				C	The number of bit for calculation	A	Setting value for UARTDR	

Downloaded from Arrow.com.

Table 24-8 List of API

Address	Contents	Using Stack (Note2)	Work register (Note1)	Argument		Return value		
				Register	Setting value	Register	Contents	
0x101A (.BTRead)	Write 128 consecutive bytes (one Page) stored in the flash memory to RAM	9bytes	WA BC DE IX	WA	Beginning address of Page to be copied (M, L)	A	0xFF	Normal end
				C	Beginning address of Page to be copied (H)		0x00	Enable Code error
				DE	Beginning address of copy destination in RAM	-	-	-
				(SP-)	0xD5 (Enable Code)			
0x1020 (.BTErsSP)	Disabling the Security Program	6bytes	WA BC DE IX	A	0xD5 (Enable Code)	A	0xFF	Normal error
							0x01	NO BUSY error
0x1022 (.BTUpdSD)	Updating the contents of shadow RAM	2bytes	WA	WA	Return address (Note3)	-	-	-

Note 1: Since working registers are rewritten in the support program, if necessary, save the contents of the work registers before calling the support program.

Note 2: While the support program is being executed, a maximum 19 bytes are used as stack (not including stack used by interrupts). Therefore, be sure to reserve a stack area beforehand.

Note 3: If 0xFFFF is set, it returns to the 2-byte address stack in SP, regardless of the value of WA.

24.5.1 .BTRead

This API reads 128-byte data after the beginning address of Page in the flash memory specified by the C register and the WA register, and write the data into the 128 bytes after RAM address specified by the DE register. Enable Code (0xD5) must be set preliminarily in (SP-). Before read operation, the lower seven bits of the A register are masked to be "0". This API can access the data (one Page) of arbitrary address in the flash memory via RAM by using with .BTWrite.

If the process is completed successfully, a return value is "0xFF". If the data set to (SP-) is not "0xD5", "0x00" is returned as a return value which indicates an Enable Code error. If this value is returned, an operation to read the flash memory and a operation to write data to RAM are not performed. After the completion of the API, FLSCR1<FAREA> is initialized to "0y00" regardless of the operation result; normal end or error.

24.5.2 .BTEraseSec

This API erases the Sector specified on the A register. Enable Code (0xD5) must be preliminarily set in the C register. Table 24-9 shows the sectors to be erased.

After erasing a sector, if the flash memory does not become BUSY state (in the case "0xD5" is not set to the C register, etc), the API returns the A register with the contents of "0x01" (NO BUSY error) as return value.

If the process is completed successfully, a return value is "0xFF".

Table 24-9 Sector to be erased

A register	Erasing sector	Erasing area
0x04 - 0x1F	Erase AREA D0	0x01000 through 0x07FFF
0x20 - 0x3F	Erase AREA D1	0x08000 through 0x0FFFF
0x40 - 0x5F	Erase AREA C0	0x10000 through 0x17FFF
0x60 - 0x7F	Erase AREA C1	0x18000 through 0x1FFFF

Downloaded from Arrow.com.

24.5.3 .BTEraseChip

All flash memory area is erased. Enable Code (0xD5) must be preliminarily set in the A register. After performing the erase operation, if the flash memory does not become BUSY state (in the case "0xD5" is not set to the A register, etc.), the API returns the A register with the contents of "0x01" (NO BUSY error) as return value.

If the process is completed successfully, a return value is "0xFF".

24.5.4 .BTGetSP

The security status of the flash memory can be obtained. The API returns the A register with the contents of (0xFF7F) in the Product ID as return value.

24.5.5 .BTSetSP

This API set the security program of the flash memory. 0x8000 must be set on the WA register and 0x0080 must be set on the DE register preliminarily. If the value set to the C register is "0xFF", the expected operation is not performed, and "0xFF" is returned to the A register.

After disabling the security program, if the flash memory does not become BUSY (in the case "0xD5" is not set to (SP-), etc.), "0x01" (NO BUSY error) is returned to the register A.

Also, the setting address exceeds 0xFFFF due to the setting of the start address and the increment address, "0x02" (address overflow error) is returned to the A register.

Setting values of C register

7	6	5	4	3	2	1	0
"1"	"1"	"1"	"1"	"1"	SB2	SB1	SB0

SB2	The write protection setting in MCU mode and serial PROM mode.	0: 1:	Enable a write protection –
SB1, SB0	The read/write protection setting in Parallel PROM mode	00: 11:	Enable a read/write protection –

Note 1: Make sure to set "1" to the bit 7, 6, 5, 4 and 3 of the C register.

Note 2: Security program is not be disabled even if "1" is set to the bit that a value is already "0".

Note 3: Set "1" to the bits that the security program is not enabled.

Note 4: Make sure not to set "0" to the bits that a value is already "0". For example, if SB1 and SB0 are already "0", and you would like to clear the SB2 from "1" to "0", set "1" to the bit 1 and bit 0, and clear "0" to the bit 2.

24.5.6 .BTErsSP

This API disables Security program for the flash memory. Enable Code (0xD5) must be preliminarily set to the A register. After disabling the security program, if the flash memory does not become BUSY state (in the case "0xD5" is not set to the A register, etc.), "0x01" (NO BUSY error) is returned to the A register.

If the process is completed successfully, a return value is "0xFF".

24.5.7 .BTConvAdr

This API converts the address from 0x01000 through 0x1FFFF specified in the C register and the WA register to the address 0x08000 through 0x0FFFF, writes the data in the WA register, and then set FLSCR1<FAREA>.

Downloaded from Arrow.com.

24.5.8 .BTCalcUART

This API calculates the proper setting for baud rate of UART from the value of the C register and the WA register. A count value captured by the 16-bit timer counter in the pulse width measurement mode is set as the WA register. Be sure to select 2/fcgck as the source clock for 16-bit timer and capture the 8-bit pulse. Set 0x08 to the C register.

RXD pin can be used as TCA pin. To capture the value, set the RXD pin as TCA pin using SERSEL<TCA0SEL> temporarily. After capturing, be sure to resume the pin to RXD pin.

The possible value range for the WA register is from 0x0020 through 0x3BFF. If the contents of the WA register is out of the range, the API returns the WA register with the contents of 0xFFFF as return value. If the conversion is made properly, the value of UARTCR2<RTSEL> is returned to the bits 5, 4 and 3 of the W register, and UARTDR to the A register as return values. The API sets the bits other than 5, 4 and 3 to "0". Therefore, set the proper value for UARTCR2<RXDNC, STOPBR>.

Note 1: If the captured value of the WA register is so small (in the range of baud rates cannot be generated by UART), an error of return values becomes large and the appropriate baud rate may not be generated.

Note 2: When the internal high-frequency oscillation (fosc) is used, the maximum baud rate calculated by the API is as shown in Table 24-10.

Table 24-10 Maximum baud rate with internal oscillation (fosc)

Clock gear [Hz]	Maximum baud rate [bps]
fcgck = fosc / 1	76800
fcgck = fosc / 2	38400
fcgck = fosc / 4	19200

The following procedure shows an example of how to calculate the transfer clock of UART in MCU mode by using the support program of BOOTROM (API).

1. Enable the timer counter input to be also used as RXD input by using the serial interface selection control register SERSEL<TCA0SEL>.
2. Set 16-bit timer counter to pulse pulse width measurement mode. And set falling-edge/L level as an external trigger and select fcgck/2 as the source clock.
3. Receive fixed data (0x80) from a master device via RXD pin. It is not necessary to enable UART.
4. Write the captured value into the WA register. Since the registers (DE, BC and IX) are rewritten in the support program, the contents of these registers should be saved before calling the support program.
5. Set the interrupt master enable flag to "disable" (DI) (IMF ← "0").
6. Set "0xD5" on FLSCR2 after setting FLSCR1<BAREA> to "1".
7. Set "0x08"(8-bit length) to the C register.
8. Call .BTCalcUART(0x101E).
9. Set bits 5, 4 and 3 of the W regiser to UARTCR2<RTSEL> and store the A register to UARTDR. If the value of the WA register is 0xFFFF which indicates an error of calculation, retry the execution from receiving the data (0x80).
10. Set "0xD5" to FLSCR2 after setting FLSCR1<BAREA> to "0".

Downloaded from Arrow.com.

Note 1: If registers (WA,BC,DE and IX register) is used in non-maskable interrupt subroutine, occurring of non-maskable interrupt may cause unexpected result.

Note 2: With success of calculation, this API returns "0" into the W register, 7, 6, 2, 1 and 0, as return values. Therefore, set proper value for these bits to select noise cancel time and receive stop bit length.

(Example) Capture the low width of 8 bit value via RXD0 pin using the 16-bit timer counter (TCA0) which is set to the pulse width measurement mode. And calculate baud rate for UART from the captured value.

```
                .BTCalcUART  equ 0x101E          ; Calculate baud rate for UART

CalcUART secion code abs = 0x1F000
; #### Set timer counter input to be also used as UART receive input ####
                LD           (SERSEL),0x40       ; Switch to TCA0 input
; #### Receive data (0x80) from a master device ####
                LD           (TA0MOD), 0x5E      ; Set the pulse width measurement mode and select falling edge/L level for
                                                 ; external trigger
                                                 ; Select fcgck/2 as source clock
sTimerStart:    LD           (TA0CR), 0x01       ; Timer start
                :            :
                Receives fixed data (0x80) via TCA0 pin.
                :            :
                LD           WA,(TA0DRL)         ; Write the captured data into the WA register
                LD           (TA0CR), 0x00       ; Timer stop
                DI
; #### Allocate BOOTROM to the data/code area ####
                LD           (FLSCR1),0x50       ; Set BAREA to "1"
                LD           (FLSCR2),0xD5       ; Reflect the FLSCR1 setting
; #### Calculation for UART setting (API) ####
                LD           C,0x08              ; The number of bit length (8 bit)
                CALL         (.BTCalcUART)       ; Calculate the UART setting data
                CMP          W, 0xFF
                J            Z, sTimerStart      ; If the value of the W register is "0xFF", retry the execution of ; receiving
                                                 ; data.
; #### Set the calculated result to UART control register ####
                LD           (UARTCR2),W         ; Set RTSEL
                LD           (UARTDR), A         ; Set baud rate register
; #### End process ####
                LD           (FLSCR1),0x40       ; Set BAREA to "0"
                LD           (FLSCR2),0xD5       ; Reflect the FLSCR1 setting
```

24.5.9 .BTUpdSD

Update the contents of the shadow RAM. The address to return to must be preliminarily set in the WA register. After updating the shadow RAM, it returns to the address specified in the WA register. However, if 0xFFFF is set to the WA register, it returns to the 2-byte address (next address of CALL instruction) stack in SP. If you would like to call this API with CALL instruction and start the next instruction after updating the shadow RAM, set 0xFFFF to the WA register.

Downloaded from Arrow.com.

(Example) Update the shadow RAM, and run a program from 0x1F900

```
                .BTUpdSD      equ 1022

UpdSD secion code abs = 0x1F800
                    :                    :
                LD            WA,0xF900
                CALL          (.BTUpdSD)
```


Downloaded from Arrow.com.

Downloaded from Arrow.com.

25. Shadow RAM

25.1 Configuration

Shadow RAM is mapped to addresses 0x0FC00 to 0x0FFFF (1024 bytes) in the data area and 0x1F800 to 0x1FFFF (2048 bytes) in the code area in the memory map. Upon reset, the contents of the Flash memory at these locations are automatically copied into the shadow RAM. After reset release, when a read or fetch instruction is executed on these locations, data is read from the shadow RAM.

When shadow RAM is not mapped, the above locations can be mapped as data RAM (3072 bytes at 0x01000 to 0x01BFF) separately from the internal RAM (3072bytes at 0x00040 to 0x00C3F) by setting the appropriate register. It is also possible to map neither shadow RAM nor data RAM.

Figure 25-1 shows the configuration of the shadow RAM control circuits.

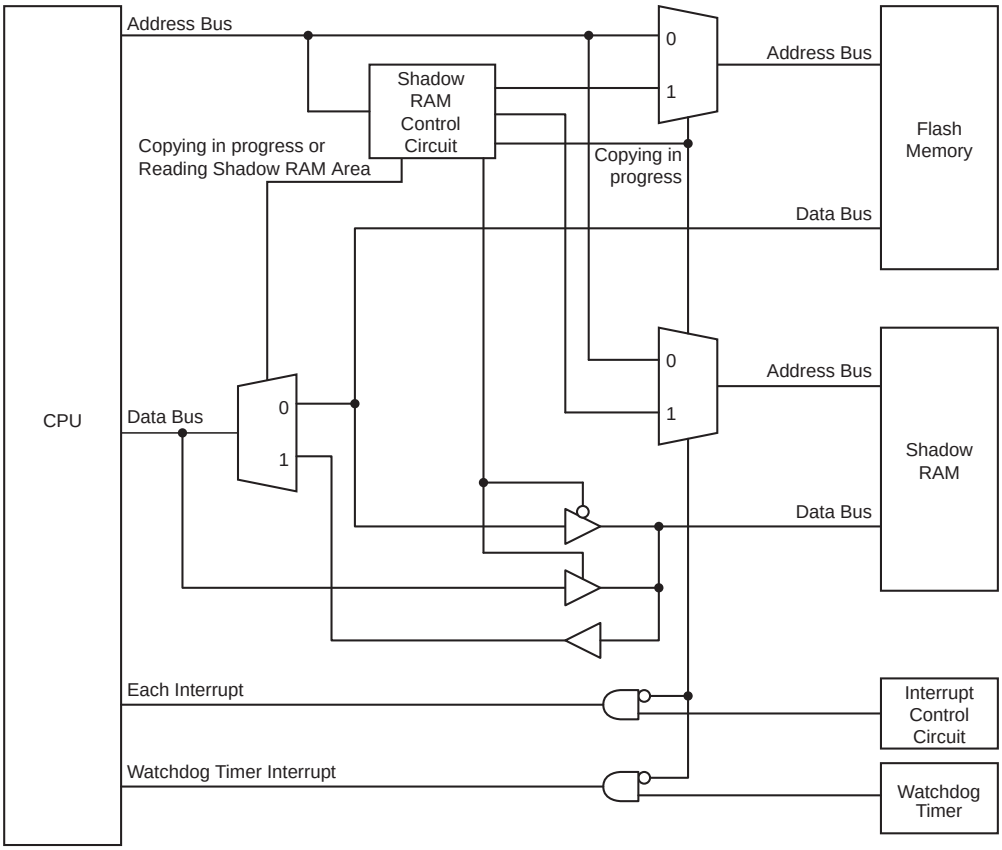


Figure 25-1 Configuration of the Shadow RAM Control Circuits

Downloaded from Arrow.com.

25.2 Control

The shadow RAM control circuits are controlled by the shadow RAM control registers 1 and 2 (SDWCR1 and SDWCR2).

Shadow RAM control register 1

SDWCR1 (0x00F7C)

	7	6	5	4	3	2	1	0
Bit Symbol	FLSOFF	FLSWUE	DADIS	DAWREN	EXPRAM	SDWDIS	SDWBSY	SDWCPY
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R	W
After reset	0	1	0	0	0	0	0	0

FLSOFF	Flash memory power control	0 :	Connect the Flash memory power supply		
		1 :	Disconnect the Flash memory power supply		
FLSWUE	Flash memory power warm-up end flag	0 :	Warming up or the power supply disconnected		
		1 :	Warm-up complete or the power supply connected		
DADIS	Shadow RAM mapping control in the data area		SDWDIS="0"		SDWDIS="1"
		0 :	Enable mapping of shadow RAM in the data area		–
		1 :	Disable mapping of shadow RAM in the data area		–
DAWREN	Shadow RAM write control in the data area		WSDWDIS="0"		SDWDIS="1"
			DADIS="0"	DADIS="1"	
		0 :	Disable writing to shadow RAM in the data area	–	
		1 :	Enable writing to shadow RAM in the data area	–	
EXPRAM	Data RAM mapping control		SDWDIS="0"		SDWDIS="1"
		0 :	–		Disable mapping of data RAM at 0x01000 to 0x01BFF
		1 :	–		Enable mapping of data RAM at 0x01000 to 0x01BFF
SDWDIS	Shadow RAM mapping control	0 :	Enable mapping of shadow RAM		
		1 :	Disable mapping of shadow RAM		
SDWBSY	Copy flag	0 :	Copying complete		
		1 :	Copying in progress		
SDWCPY	Start copying the contents of Flash memory into shadow RAM	0 :	–		
		1 :	Start copying		

Note 1: Bit 0 of SDWCR1 is read as "0".

Note 2: SDWCR1<FLSOFF> and <SDWDIS> are double-buffered. The values written to these bits are reflected in the shift register by writing 0x3B to the SDWCR2 register. In other words, the settings of SDWCR1<FLSOFF> and <SDWDIS> do not take effect until 0x3B is written to the SDWCR2 register.

Downloaded from Arrow.com.

Shadow RAM control register 2

SDWCR2 (0x00F7D)

	7	6	5	4	3	2	1	0
Bit Symbol	SDWCMD							
Read/Write	Write only							
After reset	0	0	0	0	0	0	0	0

SDWCMD	SDWCR1<FLSOFF>< SDWDIS> setting enable	0x3B : Other	Enable the values written to SDWCR1<FLSOFF> and <SDWDIS> Reserved

Note 1: SDWCR2 is read as 0x00.

Note 2: A value newly written to SDWCR1<FLSOFF> takes effect only when an instruction to write 0x3B to SDWCR2 is executed in one of the following areas. If an instruction to write 0x3B to SDWCR2 is executed in any other area, a value newly written to SDWCR1<FLSOFF> cannot be enabled and the setting of SDWCR1<FLSOFF> remains unchanged.

- BOOTROM
- RAM area (0x00040 to 0x00C3F) when RAREA = "1"
- Shadow RAM area

Note 3: A value newly written to SDWCR1<SDWDIS> takes effect only when an instruction to write 0x3B to SDWCR2 is executed in one of the following areas. If an instruction to write 0x3B to SDWCR2 is executed in any other area, a value newly written to SDWCR1<SDWDIS> cannot be enabled and the setting of SDWCR1 <SDWDIS> remains unchanged.

- BOOTROM
- RAM area (0x00040 to 0x00C3F) when RAREA = "1"
- Flash memory area

Downloaded from Arrow.com.

25.3 Memory Map

The following shows how 3072 Kbytes of RAM are mapped depending on the settings of SDWCR1<SDWDIS>, <DADIS> and < EXPRAM>.

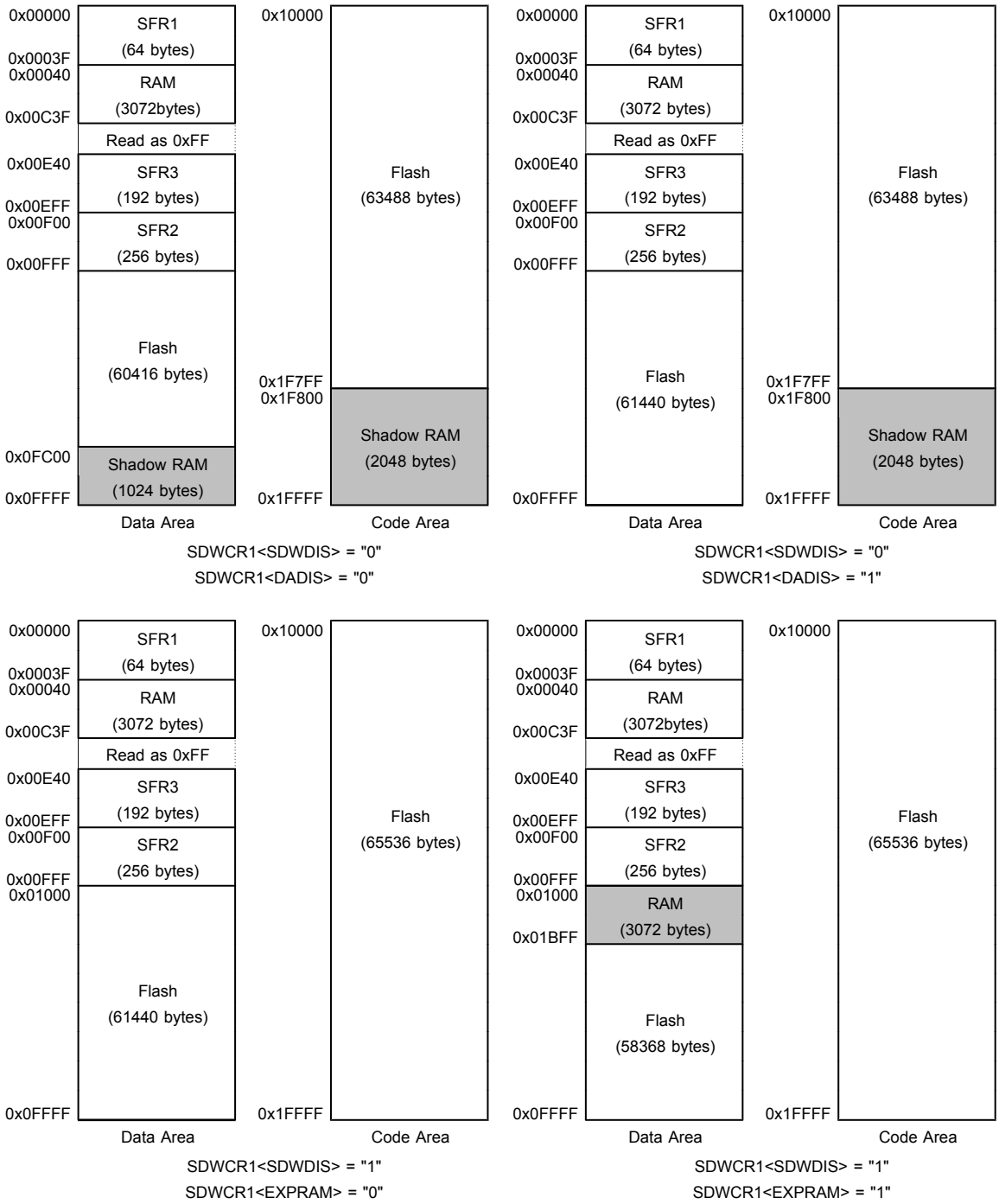


Figure 25-2 Memory Map

Downloaded from Arrow.com.

25.4 Functions

25.4.1 Copying the Flash memory

Upon reset, the contents of addresses 0x0FC00 to 0x0FFFF (1024 bytes) in the data area and 0x1F800 to 0x1FFFF (2048 bytes) in the code area of the Flash memory are automatically copied into the shadow RAM (Figure 25-3).

After reset release, if the contents of the shadow RAM are rewritten or need to be refreshed, the contents of the Flash memory can be copied into the shadow RAM by setting SDWCR1<SDWCPY> to "1".

If an interrupt occurs while the contents of the Flash memory are being copied into the shadow RAM, the operation varies depending on the interrupt source as shown in Table 25-1.

Table 25-1 Interrupt Operation During Copying

Interrupt source	Operation when an interrupt occurs while the Flash contents are being copied
Software interrupt (INTSWI) Undefined instruction interrupt (INTUNDEF)	An interrupt is accepted during copying. However, because the vector area in the shadow RAM cannot be read during copying, the vector area must be mapped in the internal RAM in advance through the setting of SYSCR3<RVCTR>.
Watchdog timer interrupt (INTWDT)	An interrupt is ignored. Because an interrupt request is not latched, no INTWDT interrupt occurs upon completion of copying. When the up counter of the watchdog timer overflows, it is initialized to "0" and continues counting. (Since the timeout period of the watchdog timer is much longer than the copying time, clearing the watchdog timer immediately before the start of copying normally prevents any INTWDT interrupt during copying.)
Other maskable interrupts	An interrupt is not accepted even when IMF (interrupt master enable flag) is set to "1". However, when IMF is set to "1", an interrupt request is latched and it is serviced upon completion of copying.

Note 1: While the contents of the Flash memory are being copied (SDWCR1<SDWBSY> = "1"), no access is allowed to the Flash memory and shadow RAM. During that period, use API subroutines of the BOOT-ROM or place the program in the RAM area (0x00040 to 0x00C3F) and wait there until copying is completed.

Note 2: While the contents of the Flash memory are being copied (SDWCR1<SDWCPY> = "1"), a read or fetch from the Flash memory or shadow RAM returns 0xFF.

Note 3: While the contents of the Flash memory are being copied (SDWCR1<SDWCPY> = "1"), do not activate IDLE1, IDLE 2 or SLEEP1 mode. Because no maskable interrupt can be accepted during copying, there is no means of exiting IDLE1, IDLE2 or SLEEP1 mode to return to NORMAL or SLOW mode.

Note 4: Before copying the contents of the Flash memory, make sure that the Flash memory is properly powered (SDWCR1<FLSOFF>="0" and SDWCR1<FLSWUE>="1"). If copying is started before the power supply has warmed up or the Flash memory is powered off during copying, the contents of the Flash memory cannot be copied properly.

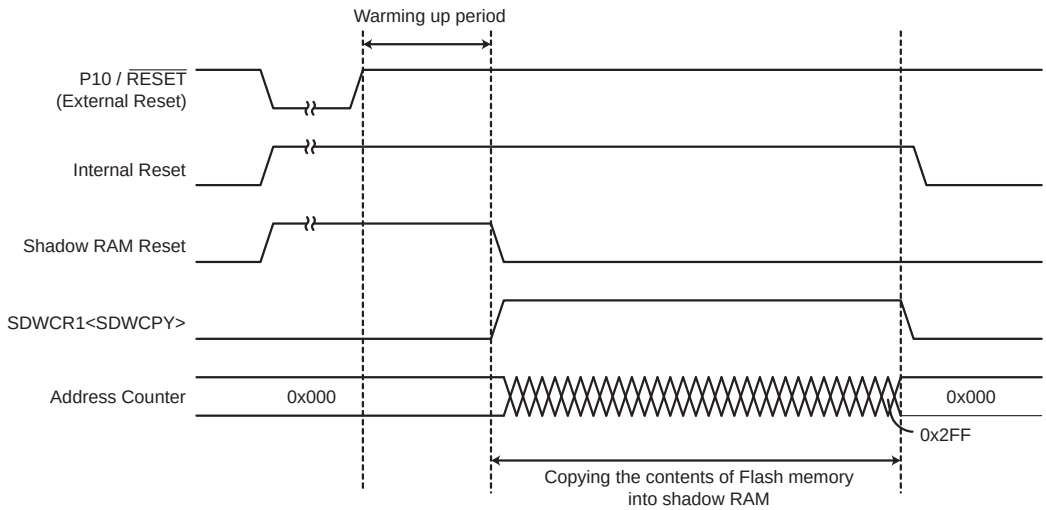


Figure 25-3 Copy Operation Timing at Reset

Downloaded from Arrow.com.

The time required for copying the contents of the Flash memory into the shadow RAM is expressed by the following equation:

tcpy = 3072 / (fcgck × 4) [s]

After SDWCR1<SDWCPY> is set to "1", whether or not copying is completed can be checked by monitoring SDWCR1<SDWBSY>.

Downloaded from Arrow.com.

(Example) Transfer the shadow RAM copy program to the internal RAM (0x00200-) to update the shadow RAM. If a nonmaskable interrupt (INTSWI or INTUNDEF) occurs while the shadow RAM is being updated, generate a system clock reset.

```
                cRAMStartAdd equ 0x0200          ; RAM start address
ShRam section code abs = 0x1F000
; #### Set nonmaskable interrupt vector in RAM area ####
                LD      HL,0x01FC                ;Set interrupt vector for INTUNDEF and INTSWI
                LDW     (HL),sINTSWI - sRAMprogStart + cRAMStartAdd
; ####Transfer code to RAM ####
                LD      HL,cRAMStartAdd
                LD      IX,sRAMprogStart
sRAMLOOP:       LD      A,(IX)                   ; Transfer code from sRAMprogStart through sRAMprogEnd
                LD      (HL),A                   ; to cRAMStartAdd
                INC     HL
                INC     IX
                CMP     IX,sRAMprogEnd
                J       NZ,code_addr(sRAMLOOP)
; #### Allocate RAM to code area and switch vector area to RAM ####
                LD      (SYSCR3),0x06            ; Set RAREA = "1" and RVCTR = "1"
                LD      (SYSCR4),0xD4            ; Enable code
; #### Update shadow RAM ####
                CALL    cRAMStartAdd             ;Call the code in RAM
; ####Next main program execution ####
                :       :                        ; Main program execution
                J       code_addr(XXXXX)
RamExe section code abs = 0x2000
; #### Code to be executed in RAM ####
sRAMprogStart:
                LD      (SDWCR1), 0x01           ; Start copying the contents of Flash into shadow RAM
sCPLOOP1:       TEST    (SDWCR1).1               ; Wait until copying is completed
                J       F, sCPLOOP1              ;
                RET                              ; Return to the main routine
; Interrupt service routine
sINTSWI:
```

Error processing execution

```
sCPLOOP2:       TEST    (SDWCR1).1               ; Wait until copying is completed
                J       F, sCPLOOP2              ;
                LD      (SYSCR2),0x10            ; Generate a system clock reset
                RETN
sRAMprogEnd:    NOP
```


Downloaded from Arrow.com.

25.4.2 Powering off the Flash memory (SDWCR1<FLSOFF>)

The power supply of the Flash memory can be turned off by setting SDWCR1<FLSOFF> to "1" and writing 0x3B to SDWCR2. The power supply of the Flash memory can be turned on by setting SDWCR1<FLSOFF> to "0" and writing 0x3B to SDWCR2.

After turning on the power supply of the Flash memory, it is necessary to wait until the Flash memory has warmed up. SDWCR1<FLSWUE> should be monitored to determine whether or not the warm-up is completed. The warm-up period is 2 to 4 ms regardless of the operating frequency.

Note 1: The power supply of the Flash memory can be turned off only when the program is executing in the BOOT ROM, RAM (0x00040 to 0x00C3F) or shadow RAM area. If SDWCR1 and SDWCR2 are set in any other area, those settings are ignored.

When an instruction to write 0x3B to SDWCR2 is executed, the area in which the program resides is checked. If the program is executing from an area other than the BOOTROM, RAM or shadow RAM, SDWCR1<FLSOFF> is cleared to "0".

Note 2: After the power supply of the Flash memory is turned off, a read from the Flash memory area returns 0xFF. A fetch from the Flash memory area causes a software interrupt.

(Example) Turn off the power supply of the Flash memory from the shadow RAM and switch the operating mode from NORMAL2 to SLOW1 mode. Then, after detecting H level on port P70, turn on the power supply of the Flash memory, switch the operating mode from SLOW1 to NORMAL2 mode, and return to the main routine.

```
ShRam section code abs = 0x1F800
slow_task:   LD      (SDWCR1), 0x80      ; Turn off the power supply of the Flash memory
             LD      (SDWCR2), 0x3B
             SET     (SYSCR1).4          ; SYSCR2<SYSCK> = 1 (switch to SLOW2 mode)
                                         ; (Switch the system clock to low-frequency reference clock)
             NOP                         ; Wait for 2 machine cycles
             NOP
             CLR     (SYSCR2).6          ; SYSCR2<XEN> = 0 (Switch to SLOW1 mode)
sLOOP1:
```

Program execution

```
             TEST    (P7PRD).0           ; Poll for rising edge of port P70
             J       T, code_addr(sLOOP1)
             LD      (SDWCR1), 0x00      ; Turn on the power supply of the Flash memory
             LD      (SDWCR2), 0x3B
sLOOP2:      TEST    (SDWCR1).6          ; Wait until the Flash memory has warmed up
             J       T, code_addr(sLOOP2)
             LD      (WUCCR), 0x09
             LD      (WUCDR), 0x9D       ; Set the warm-up period
             SET     (SYSCR2) .6         ; SYSCR2<XEN> = 1 (Switch to SLOW2 mode)
sLOOP3:      TEST    (ILL).4             ; Wait until the high-frequency clock has warmed up
             J       T, code_addr(sLOOP3)
             LD      (ILL), 0xEF         ; Clear IL4
             CLR     (SYSCR1).4          ; SYSCR2<SYSCK> = 0 (Switch to NORMAL2 mode)
             RET                         ; Return to the main routine
```


Downloaded from Arrow.com.

25.4.3 Shadow RAM mapping control in the data area (SDWCR1<DADIS>)

When SDWCR1<SDWDIS> = "0", setting SDWCR1<DADIS> to "1" disables the mapping of shadow RAM to addresses 0x0FC00 to 0x0FFFF in the data area and maps Flash memory at these locations. Clearing SDWCR1<DADIS> to "0" enables the mapping of shadow RAM in the data area.

In executing erase and write operations on the Flash memory (including API operations of the BOOT-ROM), command sequences or verify operations cannot be performed on addresses 0x0FC00 to 0x0FFFF when SDWCR1 <DADIS> is cleared to "0". Therefore, to erase or program the Flash memory, SDWCR1<DADIS> must be set to "1" to disable the mapping of shadow RAM in the data area.

25.4.4 Shadow RAM mapping control (SDWCR1<SDWDIS>)

When SDWCR1<SDWDIS> is set to "1" and 0x3B is written to SDWCR2, the mapping of shadow RAM (0x0FC00 to 0x0FFFF in the data area and 0x1F800 to 0x1FFFF in the code area) is disabled and Flash memory is mapped to these locations. When SDWCR1<SDWDIS> is cleared to "0" and 0x3B is written to SDWCR2, shadow RAM is mapped to these locations, instead of Flash memory.

25.4.5 Shadow RAM write control in the data area (SDWCR1<DAWREN>)

When SDWCR1<SDWDIS> = "0" and SDWCR1<DADIS> ="0", setting SDWCR1<DAWREN> to "1" allows the CPU to write to the shadow RAM mapped in the data area. Clearing SDWCR1<DAWREN> to "0" disables writes to the shadow RAM in the data area.

Note that clearing SDWCR1<DAWREN> to "0" does not automatically replace the contents of the shadow RAM with the contents of the Flash memory. If the contents of the shadow RAM have been rewritten, discrepancies arise between the contents of the shadow RAM and those of the Flash memory. If the contents of the shadow RAM need to match those of the Flash memory, the contents of the Flash memory must be explicitly copied into the shadow RAM (see "25.4.1 Copying the Flash memory").

25.4.6 Data RAM mapping control (SDWCR1<EXPRAM>)

After the mapping of shadow RAM is disabled, setting SDWCR1<EXPRAM> to "1" makes 3072 bytes at addresses 0x01000 to 0x01BFF to be mapped as data RAM, instead of Flash memory, separately from the internal RAM (3072bytes at 0x00040 to 0x00C3F).

When used as data RAM, these locations are not affected by SYSCR3<RAREA> (cannot be mapped in the code area). In this case, the Flash memory cannot be accessed at these locations in the data area.

Downloaded from Arrow.com.

Downloaded from Arrow.com.

26. Serial PROM Mode

26.1 Outline

The TMP89FW24A has a 4-Kbyte BOOTROM (mask ROM) for programming flash memory. The BOOT-ROM is available in serial PROM mode. The Serial PROM mode is controlled by the RXD0 / SI0, TXD0 / SO0, MODE and $\overline{\text{RESET}}$ pins. In serial PROM mode, communication is performed via the UART or SIO.

Table 26-1 Operating Range in Serial PROM Mode

Parameter	Min	Max	Unit
Power supply voltage	2.7	5.5	V
High frequency	1	16	MHz

Note 1: High frequency in the above table means that external oscillator is used.

Note 2: Serial PROM mode can be used even if external oscillator is not connected to external pin, because system clock is started by internal high frequency after reset. When the clock change command is used, system clock can be changed to external clock.

26.2 Security

In serial PROM mode, two security functions are provided to prevent illegal memory access attempts by a third party: password protection and Security Program. For more security-related information, refer to "26.12 Security".

Downloaded from Arrow.com.

26.3 Serial PROM Mode Setting

26.3.1 Serial PROM mode control pins

To execute on-board programming, start the device in serial PROM mode. Table 26-2 shows the pin settings for starting the device in serial PROM mode.

Table 26-2 Serial PROM Mode Setting

Pin	Setting
RXD0 / SI0 / P21 pin	H level
TXD0 / SO0 / P20 pin	H level
MODE and $\overline{\text{RESET}}$ pins	↗

Note: The RXD0/SI0/P21 and TXD0/SO0/P20 pins must be pulled up to the H level until the device starts in serial PROM mode.

Table 26-3 Pin Functions in Serial PROM Mode

Pin name (in serial PROM mode)	Input/output	Function		Pin name (in MCU mode)
TXD0 / SO0	Output	Serial PROM mode control/serial data output	(Note 1)	TXD0 / SO0 / P20 / SEG37
RXD0 / SI0	Input	Serial PROM mode control/serial data input		RXD0 / SI0 / P21 / SEG36
$\overline{\text{RESET}}$	Input	Serial PROM mode control		$\overline{\text{RESET}}$
MODE	Input	Serial PROM mode control		MODE
SCLK0	Input	Serial clock input (if SIO is used) This pin is in the high-impedance state in serial PROM mode. If the UART is used, the port input is physically fixed to a specified input level in order to prevent a penetration current. To enable the port input, the SPCR<PIN1> must be set to "1" by the RAM loader control program.		SCLK0
VDD	Power supply	2.7 V to 5.5 V		
AVDD	Power supply	Connect to VDD.		
VSS	Power supply	0 V		
AVSS	Power supply	Connect to VSS.		
VAREF	power supply	Leave open or apply reference voltage.		
Ports other than RXD0 and TXD0	Input/ output	These ports are in the high-impedance state in serial PROM mode. The port input is physically fixed to a specified input level in order to prevent a penetration current (the port input is disabled). To enable the port input, the SPCR<PIN0> must be set to "1" by the RAM loader ocntrol program.		
COM0 ~ COM3	Output	The output is at the L level in serial PROM mode.		
VLC	Power supply	Connect to VDD or apply an LCD drive voltage.		
XIN	Input	Connect a resonator to make these pins self-oscillate.		
XOUT	Output			

Note 1: If other parts are mounted on a user board, they may interfere with data being communicated through these communication pins during on-board programming. It is recommended that these parts be somewhat isolated to prevent the pins from being affected.

Downloaded from Arrow.com.

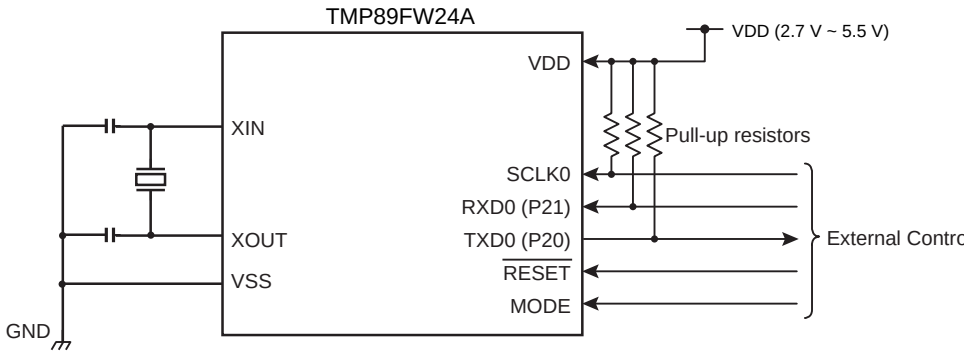


Figure 26-1 Serial PROM Mode Pin Setting

Note 1: When using the UART, there is no need to control the SCLK0 pin.

Note 2: For information on other pin settings, refer to "Table 26-3 Pin Functions in Serial PROM Mode".

Downloaded from Arrow.com.

26.4 Examples of Connections for On-Board Programming

Figure 26-2 shows examples of connections for on-board programming.

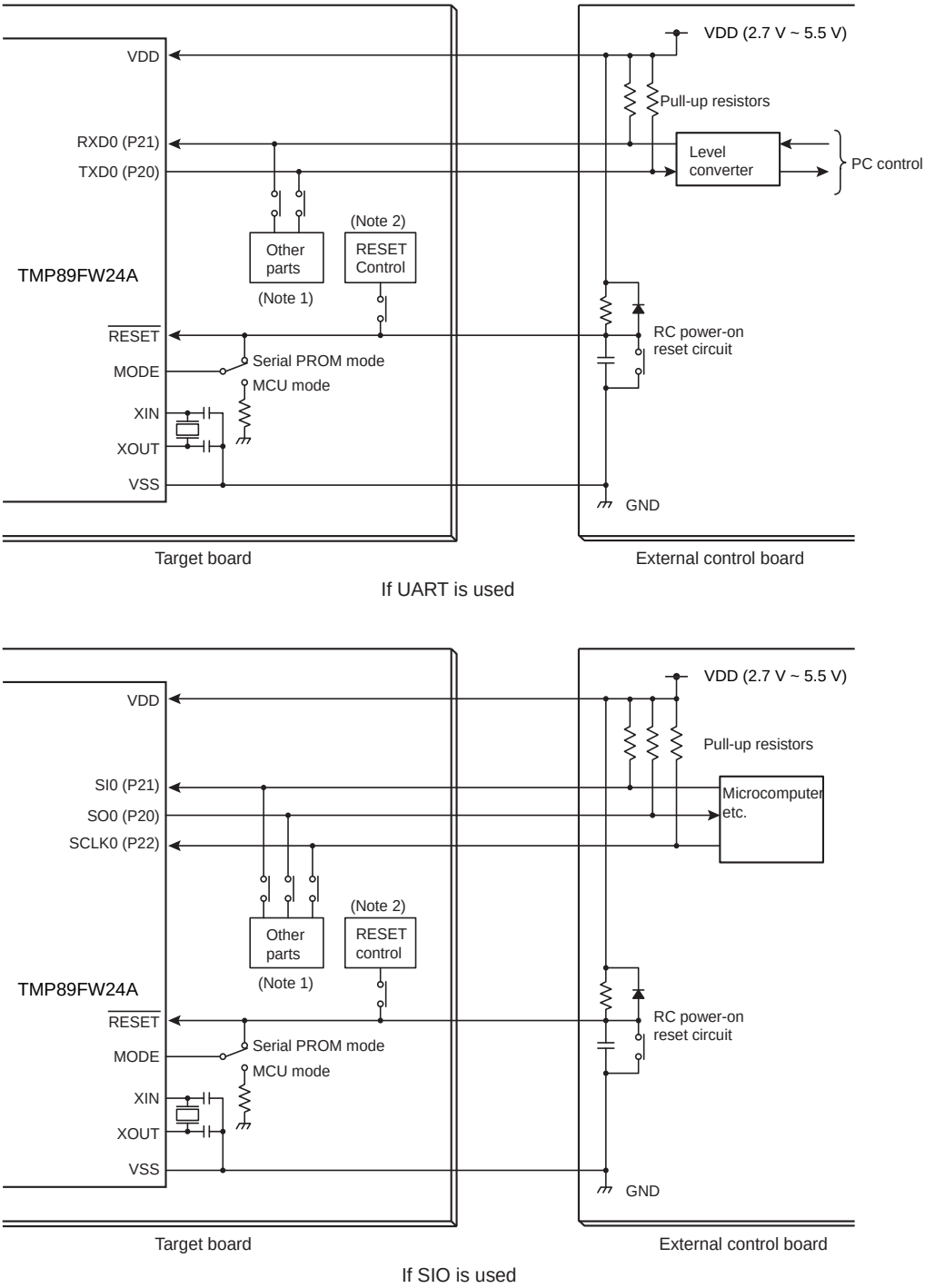


Figure 26-2 Examples of Connections for On-Board Programming

Note 1: If other parts on a target board interfere with the UART communication in serial PROM mode, disconnect these pins by using a jumper or switch.

Note 2: If the reset control circuit on a target board interferes with the startup of serial PROM mode, disconnect the circuit by using a jumper, etc.

Note 3: For information on other pin settings, refer to "Table 26-3 Pin Functions in Serial PROM Mode".

Downloaded from Arrow.com.

26.5 Starting in Serial PROM Mode

To start the device in serial PROM mode, follow the steps below. For information on the detailed timing, refer to "26.14.1 Reset timing".

1. Supply power to the VDD pin.
2. Set the $\overline{\text{RESET}}$ and MODE pins to low.
3. Set the RXD0/SI0/P21 and TXD0/SO0/P20 pins to high.
4. Wait until the power supply and clock oscillation stabilize.
5. Set the $\overline{\text{RESET}}$ and MODE pins from low to high.
6. Input the matching data 0x86 or 0x30 to the RXD0/SI0/P21 pin after the setup period has elapsed.

Downloaded from Arrow.com.

26.6 Interface Specifications

The serial PROM mode supports two communication methods: UART and SIO. The communication method is selected based on the first serial data value received after a reset.

To execute on-board programming, the communication format of the external controller (personal computer, microcontroller, etc.) must be set as described below.

26.6.1 SIO Communication

- Transfer rate: 250 kbps (Max)
- Data length: 8 bits
- Slave (external clock)
- Hardware flow control (SC0 pin)

If the TMP89FW24A receives serial data "0x30" after a reset, it starts the SIO communication.

In the SIO communication, the TMP89FW24A functions as a slave device. Therefore, the external controller must supply the TMP89FW24A with serial clock pulses (via the SCLK0 pin) for synchronization.

If the TMP89FW24A is not outputting serial data, it controls the hardware flow by using the SO0 pin. If internal data processing is not completed yet, though data has been received, the SO0 pin outputs the L level. If internal data processing has progressed to a near-completion state or if it has been completed, the SC0 pin outputs the H level. The external controller must check the status of the SC0 pin before it starts to supply serial clock pulses.

26.6.2 UART Communication

- Baud rate: 9600 to 115200 bps (automatic detection)
- Data length: 8 bits (LSB first)
- Parity bit: None
- STOP bit: 1 bit

If the TMP89FW24A receives serial data "0x86" after a reset, it starts the UART communication. It also measures the pulse width of the received data (0x86), and automatically establishes the reference baud rate. In all subsequent data communication transactions, this reference baud rate is used. For information on the communication timings of each operation command, refer to "26.14 AC Characteristics (UART)".

Usable baud rates differ depending on the operating frequency and are shown in Table 26-4. However, there is the possibility of data communication not working properly, even if a baud rate shown in Table 26-4 is used, because data communication is affected by frequency errors of both a resonator and external controller (personal computer, etc.), the load capacity of a communication pin, and various other factors.

Downloaded from Arrow.com.

Table 26-4 Usable Baud Rates as a General Guideline

	9600 bps	19200 bps	38400 bps	57600 bps	115200 bps
16 MHz	O	O	O	O	O
10 MHz	O	O	O	O	O
8 MHz	O	O	O	O	O
7.3728 MHz	O	O	O	O	O
6.144 MHz	O	O	O	–	–
6 MHz	O	O	O	O	O
5 MHz	O	O	O	–	–
4.9152 MHz	O	O	O	O	–
4.19 MHz	O	O	O	–	–
4 MHz	O	O	O	O	O
2 MHz	O	O	O	O	–
1 MHz	O	O	–	–	–

Note 1: O : Can be used, – : Cannot be used.

Note 2: High frequency in the above table means that external oscillator is used. When the internal high frequency is used, baud rate should be set to 76800bps or less.

Downloaded from Arrow.com.

26.7 Memory Mapping

Figure 26-3 shows memory maps in serial PROM mode and MCU mode.

In serial PROM mode, the BOOTROM (mask ROM) is mapped to 0x1000 through 0x17FF in the data area and 0x1000 through 0x1FFF in the code area respectively.

To program or erase flash memory with a user-created program by using the RAM loader (0x60) command, you need to switch between memory areas through the flash memory control registers (FLSCR1 and FLSCR2). For information on how to specify addresses, refer to the chapter on "Flash Memory".

When the flash memory write (0x30) command or the flash memory erase (0xF0) command is executed, the BOOTROM automatically converts addresses. Therefore, the flash memory is addressed at 0x01000 through 0x1FFFF as in MCU mode (when FLSCR1<BAREA>="0").

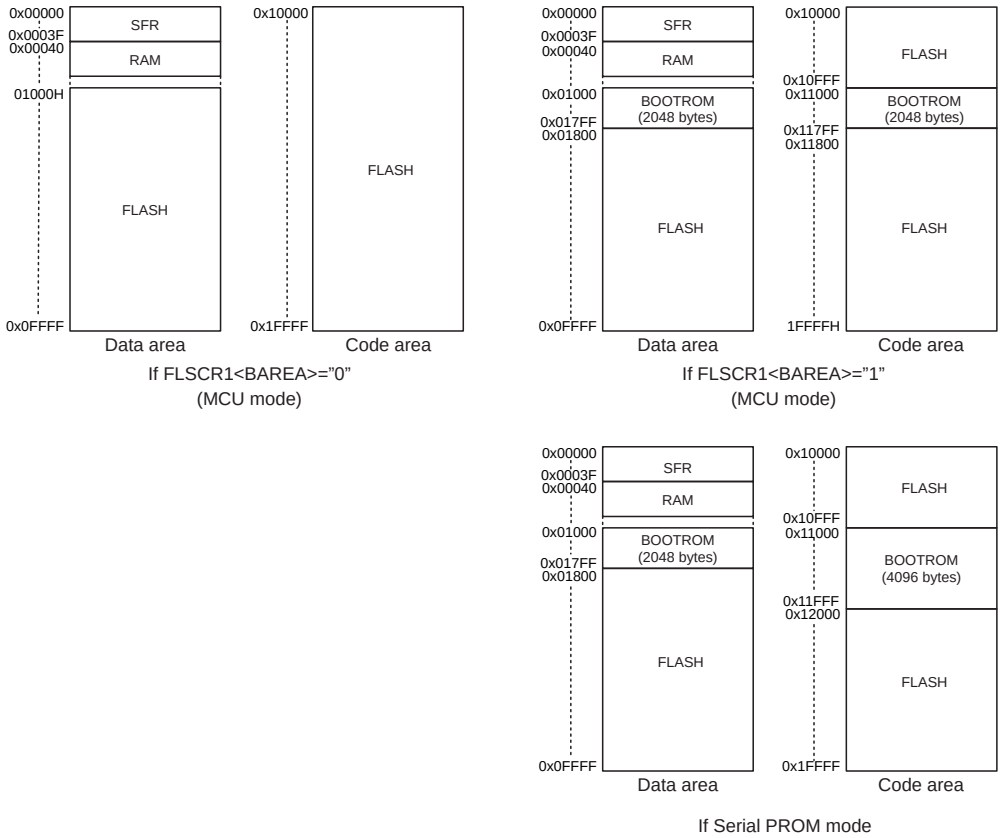


Figure 26-3 Memory Mapping

Downloaded from Arrow.com.

26.8 Operation Commands

In serial PROM mode, the commands shown in Table 26-5 are used. After a reset is released, the TMP89FW24A goes into a standby state and awaits arrival of matching data 1 (0x86 or 0x30).

Table 26-5 Operation Commands in Serial PROM Mode

Command data	Operation command	Description
0x86 or 0x30	Setup (matching data 1, 2)	After a reset is released, the serial PROM mode always starts operation with this command. If matching data 1 is 0x86, communication starts in the UART format. If matching dta 1 is 0x03, communication starts in the SIO format.
0xF0	Flash memory erase	Data in the flash memory area (addresses 0x01000 through 0x1FFFF) can be erased.
0x30	Flash memory write	Data can be written to the flash memory area (addresses 0x01000 through 0x1FFFF).
0x40	Flash memory read	Data can be read from the flash memory area (addresses 0x01000 through 0x1FFFF).
0x60	RAM loader	Data can be written to a specified RAM area (addresses 0x0060 through 0x0C3F).
0x90	Flash memory SUM output	The 2-byte checksum of the entire flash memory (addresses 0x01000 through 0x1FFFF) is output from the upper byte followed by the lower byte.
0xC0	Product ID code output	Product ID codes are output.
0xC3	Flash memory status output	The security program status and other status codes are output.
0xFA	Flash memory security setting	This command enables Security Program.
0xA0	Clock change	This command is used to change the system clock and baud rate.

Each command is outlined below. For detailed information on how each command works, refer to 26.8.1 and subsequent sections.

1. Flash memory erase command

   Either Chip Erase (erasing the entire flash memory) or Sector Erase (erasing in 32- Kbyte or 28-Kbyte units) can be used to erase the flash memory. Data in the erase area is 0xFF. When Security Program is enabled or the option code EPFC_OP is set to 0xFF, Sector Erase cannot be executed. To disable Security Program, execute Chip Erase. The TMP89FW24A performs password authentication before erasing the flash memory except when the product is blank or EPFC_OP is 0xFF. The flash memory erase command can only be executed after successful password authentication.

Note: If checksum errors occur when performing write/erase operation on the Flash memory in serial PROM mode, recheck the checksum using the Flash memory SUM output command.

2. Flash memory write command

   Data can be written in 128-Kbyte units to specified addresses in the flash memory. The external controller should transmit the data to be written as binary data in Intel Hex format. If no error occurs until the end record is reached, the TMP89FW24A calculates the checksum of the entire flash memory (0x01000 through 0x1FFFF), and returns the result. When Security Program is enabled, the flash memory write command cannot be executed. In this case, execute Chip Erase beforehand by using the flash memory erase command. The TMP89FW24A performs password authentication before executing the flash memory write command except when the product is blank. The flash memory write command can only be executed after successful password authentication.

Note: If checksum errors occur when performing write/erase operation on the Flash memory in serial PROM mode, recheck the checksum using the Flash memory SUM output command.

3. Flash memory read command

Downloaded from Arrow.com.

Data can be read from a specified address in the flash memory in single-byte units. The external controller should transmit the start address from where data is to be read and the number of bytes to be read. The TMP89FW24A outputs the specified number of bytes, then calculates the checksum and returns the result. When Security Program is enabled, the flash memory read command cannot be executed. In this case, execute Chip Erase beforehand by using the flash memory erase command. The TMP89FW24A performs password authentication before executing the flash memory read command except when the product is blank. The flash memory read command can only be executed after successful password authentication.

4. RAM loader command

The RAM loader transfers the Intel Hex format data sent by the external controller to the built-in RAM. If the transfer is completed successfully, the TMP89FW24A calculates the checksum and returns the result. Then, it jumps to the RAM address specified by the first data record and starts to execute the user program. When Security Program is enabled, the RAM loader command cannot be executed. In this case, execute Chip Erase beforehand by using the flash memory erase command. The TMP89FW24A performs password authentication before executing the RAM loader command except when the product is blank. The RAM loader command can only be executed after successful password authentication.

5. Flash memory SUM output command

This command calculates the checksum of the entire flash memory (0x01000 through 0x1FFFF) and returns the result.

6. Product ID code output command

This is a code to output used to identify a product. The output code consists of information on the ROM area and the RAM area respectively. The external controller reads this code to identify the product to which data is to be written.

7. Flash memory status output command

The status of 0x1FFE0 through 0x1FFFF and that of the security program are output. The external controller reads this code to identify the status of flash memory.

8. Flash memory security setting command

This command is used to prohibit reading from or writing to the flash memory in parallel mode and writing to the flash memory in MCU mode. In Serial PROM mode, the flash memory write, RAM loader and flash memory read commands are prohibited. To disable Security Program, execute Chip Erase by using the flash memory erase command.

9. Clock change command

This command is used to change the system clock and baud rate clock. The system clock can be selected from the internal high-frequency clock and external high-frequency clock. After the system clock is changed, the baud rate can automatically be set by inputting the reference clock from an external source.

2026-04-21
Downloaded from Arrow.com.

26.8.1 Flash memory erase command (0xF0)

Table 26-6 shows the flash memory erase command.

Table 26-6 Flash Memory Erase Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1(0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte 6th byte	Operation command data (0xF0) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0xF0) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)
	7th byte 8th byte	Password count storage address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address 15 t0 08	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : mth byte	Password string -	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	(n-2)th byte	Erase area specification	Baud rate after adjustment	-
	(n-1)th byte	-	Baud rate after adjustment	OK: Checksum (upper byte) (Note 3) Error: No data transmitted
	nth byte	-	Baud rate after adjustment	OK: Checksum (lower byte) (Note 3) Error: No data transmitted
	(n+1)th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: "0xyy × 3" means that the device goes into an idle state after transmitting 3 bytes of 0xyy.

Note 2: For information on the erase area specification, refer to "26.8.1.1 Specifying the erase area". For information on the checksum, refer to "26.10 Checksum (SUM)". For information on the password, refer to "26.12.1 Password".

Note 3: Do not transmit a password string if the flash memory has "0xFF" at address 0x1FFFA or it is blank. (However, the password count storage address and the password comparison start address must be transmitted.)

Note 4: When the flash memory has "0xFF" at address 0x1FFFA, sending a value less than 0x80 as the (n-2)th byte (i.e. executing Sector Erase) causes the TMP89FW24A to go into an idle state.

Note 5: When a password error occurs, the TMP89FW24A stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin and restart the serial PROM mode.

Note 6: If a communication error occurs during the transfer of a password address or a password string, the TMP89FW24A stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin and restart the serial PROM mode.

Downloaded from Arrow.com.

26.8.1.1 Specifying the erase area

The (n-2)th byte of the flash memory erase command specifies the area to be erased in the flash memory.

The ERASEC register specifies the address range to be erased.

If a value less than 0x80 is specified, Sector Erase (erasing in 32 or 28-Kbyte units) is executed. However, executing Sector Erase when the data at address 0x1FFFA is "0xFF" or when Security Program is enabled will cause the device to go into an infinite loop state.

If a value equal to or larger than 0x80 is specified, Chip Erase (erasing the entire flash memory) is executed. This also clears Security Program. Therefore, to disable Security Program, execute Chip Erase instead of Sector Erase.

2026-04-21
Downloaded from Arrow.com.

Erase range (data at [n-2]th byte)

7	6	5	4	3	2	1	0
ERASEC							

			TMP89FW24A erase range	Actual address	Specified by an instruction	
					FLSCR1 <FAREA>	6th address
ERASEC	Erase range	0x00	Reserved			
		0x01	Reserved			
		0x02	Reserved			
		0x03	Reserved			
		0x04	Sector Erase (partial erase) 0x01000 to 0x07FFF (AREA D0)	0x01000	01	0x9000
		~		~		~
		0x1F		0x07C00		0xFC00
		0x20	Sector Erase (partial erase) 0x08000 to 0x0FFFF (AREA D1)	0x08000	00	0x8000
		~		~		~
		0x3F		0x0FC00		0xFC00
		0x40	Sector Erase (partial erase) 0x10000 to 0x17FFF (AREA C0)	0x10000	11	0x8000
		~		~		~
		0x5F		0x17C00		0xFC00
		0x60	Sector Erase (partial erase) 0x18000 to 0x1FFFF (AREA C1)	0x18000	10	0x8000
		~		~		~
		0x7F		0x1FC00		0xFC00
		0x80 or larger	Chip Erase (erasing the entire flash memory) (After the entire flash memory is erased, Security Program is cleared.)			

Note 1: If Sector Erase is performed on a non-existent area of the flash memory, the TMP89FW24A stops communication and goes into an idle state.

Note 2: If a value indicated as reserved is transmitted, the TMP89FW24A stops communication and goes into an idle state.

Downloaded from Arrow.com.

26.8.2 Flash Memory Write Command (operation command: 0x30)

Table 26-7 shows the transfer format of the flash memory write command.

Table 26-7 Transfer Format of the Flash Memory Write Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1(0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) : Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte 6th byte	Operation command data (0x30) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x30) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)
	7th byte 8th byte	Password count storage address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error:No data transmitted
	11th byte 12th byte	Password count storage address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error:No data transmitted
	13th byte 14th byte	Password comparison start address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error:No data transmitted
	15th byte 16th byte	Password comparison start address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : mth byte	Password string (Note) -	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	(m+1)th byte : (n-3)th byte	Intel Hex format (binary)	Baud rate after adjustment	- -
	(n-2)th byte	-	Baud rate after adjustment	OK: 0x55 Overwrite detected: 0xAA
	(n-1)th byte	-	Baud rate after adjustment	OK: Checksum (upper byte) (Note 3) Error: No data transmitted
	nth byte	-	Baud rate after adjustment	OK: Checksum (lower byte) (Note 3) Error: No data transmitted
	(n+1)th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: “0xyy × 3” means that the device goes into an idle state after transmitting 3 bytes of 0xyy. For further information, refer to Table 26-18.

Note 2: For information on the Intel Hex format, refer to "26.11 Intel Hex Format (Binary)". For information on the checksum, refer to "26.10 Checksum (SUM)". For information on the password, "26.12.1 Password".

Note 3: If the area 0x1FFE0 through 0x1FFFF is all “0xFF”, password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even for a blank product. If the password count storage address and/or password comparison start address is/are incorrect, a password error occurs, the TMP89FW24A stops communication, and it goes in-

Downloaded from Arrow.com.

to an idle state. Therefore, if a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 4: If Security Program is enabled in the flash memory or if a password error occurs, the TMP89FW24A stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 5: If a communication error occurs during the transfer of a password address or a password string, the TMP89FW24A stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 6: If the entire flash memory is programmed with the same value, do not write only to addresses 0x1FFE0 to 0x1FFFF. If data is only written to these addresses, a password error occurs, and subsequent operations cannot be performed.

Note 7: The (n-2)th byte is a flag for detecting an overwrite. If addresses to be programmed contain data other than 0xFF, the TMP89FW24A sends 0xAA as the (n-2)th byte. (In this case, no write operation is performed.) The checksum to be transmitted at the (n-1)th and nth bytes is calculated from all relevant addresses regardless of whether or not they have been written. Therefore, if an overwrite is detected, the checksum of transmitted data does not match the checksum at the (n-1)th and nth bytes.

Downloaded from Arrow.com.

26.8.3 Flash memory read command (operation command: 0x40)

Table 26-10 shows the transfer format of the flash memory read command.

Table 26-8 Transfer Format of the Flash Memory Read Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1(0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte 6th byte	Operation command data (0x40) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x40) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)
	7th byte 8th byte	Password count storage address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : mth byte	Password string -	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	(m+1)th byte (m+2)th byte	Read start address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+3)th byte (m+4)th byte	Read start address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+5)th byte (m+6)th byte	Read start address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+7)th byte (m+8)th byte	Number of bytes to read 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+9)th byte (m+10)th byte	Number of bytes to read 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted

Downloaded from Arrow.com.

Table 26-9 Transfer Format of the Flash Memory Read Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	(m+11)th byte (m+12)th byte	Number of bytes to read 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+13)th byte : (n-2)th byte		Baud rate after adjustment Baud rate after adjustment	Memory data Memory data
	(n-1)th byte	-	Baud rate after adjustment	OK: Checksum (upper byte) Error: No data transmitted
	nth byte	-	Baud rate after adjustment	OK: Checksum (lower byte) Error: No data transmitted
	(n+1)th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: “0xyy × 3” means that the device goes into an idle state after transmitting 3 bytes of 0xyy. For further information, refer to Table 26-18.

Note 2: For information on the checksum, refer to "26.10 Checksum (SUM)". For information on the password, refer to "26.12.1 Password".

Note 3: If the area 0x1FFE0 through 0x1FFFF is all 0xFF, password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even for a blank product. If the password count storage address and/or password comparison start address are/is incorrect, a password error occurs; the TMP89FW24A stops communication, and goes into an idle state. Therefore, if a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 4: If Security Program is enabled in the flash memory or if a password error occurs, the TMP89FW24A stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and start the serial PROM mode.

Note 5: If a communication error occurs during the transfer of a password address or a password string, the TMP89FW24A stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 6: If the number of bytes received at the (m+7)th, (m+9)th and (m+11)th bytes is 0x000000 or exceeds the size of internal memory, the TMP89FW24A stops communication and goes into an idle state.

Downloaded from Arrow.com.

26.8.4 RAM loader command (operation command: 0x60)

Table 26-10 shows the transfer format of the RAM loader command.

Table 26-10 Transfer Format of the RAM Loader Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 (0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte 6th byte	Operation command data (0x60) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x60) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)
	7th byte 8th byte	Password count storage address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : mth byte	Password string -	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(m+1)th byte : (n-2)th byte	Intel Hex format (binary)	Baud rate after adjustment Baud rate after adjustment	- -
	(n-1)th byte	-	Baud rate after adjustment	OK: Checksum (upper byte) (Note 3) Error: No data transmitted
	nth byte	-	Baud rate after adjustment	OK: Checksum (lower byte) (Note 3) Error: No data transmitted
RAM	-	The program jumps to the start address of RAM in which the first transferred data is written, and executes itself.		

Note 1: “0xyy × 3” means that the device goes into an idle state after transmitting 3 bytes of 0xyy. For further information, refer to Table 26-18.

Note 2: For information on the Intel Hex format, refer to "26.11 Intel Hex Format (Binary)". For information on the checksum, refer to "26.10 Checksum (SUM)". For information on the password, refer to "26.12.1 Password".

Note 3: If the area 0x1FFE0 through 0x1FFFF is all 0xFF, password authentication is not performed and, therefore, the password string need not be transmitted. The password count storage address and password comparison start address, however, must be specified, even for a blank product. If the password count storage address and/or password comparison start address are/is incorrect, a password error occurs; the TMP89FW24A stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Downloaded from Arrow.com.

Note 4: After sending a password string, do not send the end record only. If the TMP89FW24A receives the end record after receiving a password string, it may malfunction.

Note 5: If Security Program is enabled in the flash memory or if a password error occurs, the TMP89FW24A stops communication and goes into an idle state. Therefore, if a password error occurs, initialize theTMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 6: If a communication error occurs during the transfer of a password address or password string, the TMP89FW24A stops communication and goes into an idle state. Therefore, when a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Downloaded from Arrow.com.

26.8.5 Flash memory SUM output command (operation command: 0x90)

Table 26-11 shows the transfer format of the flash memory SUM command.

Table 26-11 Transfer Format of the Flash Memory SUM Output Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 (0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK: Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte 6th byte	Operation command data (0x90) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x90) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)
	7th byte	-	Baud rate after adjustment	0x55 : Å| 0xAA: All data are 0xFF.
	8th byte	-	Baud rate after adjustment	OK: Checksum (upper byte) (Note 2) Error: No data transmitted
	9th byte	-	Baud rate after adjustment	OK: Checksum (lower byte) (Note 2) Error: No data transmitted
	10th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: “0xyy × 3” means that the device goes into an idle state after transmitting 3 bytes of 0xyy. For further information, refer to Table 26-18.

Note 2: For information on the checksum, refer to "26.10 Checksum (SUM)".

Note 3: If data to be included in the checksum are all 0xFF, the 7th byte becomes 0xAA. If data to be included in the checksum contain any byte that is not 0xFF, the 7th byte becomes 0x55.

Downloaded from Arrow.com.

26.8.6 Product ID code output command (operation command: 0xC0)

Table 26-12 shows the transfer format of the product ID code output command.

Table 26-12 Transfer Format of the Product ID Code Output Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller	
BOOT ROM	1st byte 2nd byte	Matching data 1(0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	-(Automatic baud rate adjustment) OK:Echo back data (0x86 or 0x30) Error: No data transmitted	
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted	
	5th byte 6th byte	Operation command data (0xC0) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0xC0) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)	
	7th byte		Baud rate after adjustment	0x3A	Start mark
	8th byte		Baud rate after adjustment	0x13	Number of transfer data (from 9th to 27th bytes)
	9th byte		Baud rate after adjustment	0x03	Length of address (3 bytes)
	10th byte		Baud rate after adjustment	0xFC	Reserved
	11th byte		Baud rate after adjustment	0x05	Reserved
	12th byte		Baud rate after adjustment	0x00	Reserved
	13th byte		Baud rate after adjustment	0x00	Reserved
	14th byte (Note 3)			0xF8	ROM size code
	15th byte		Baud rate after adjustment	0x01	First block count (1 block)
	16th byte (Note 3)		Baud rate after adjustment	0x00	First address of ROM (upper byte)
	17th byte (Note 3)		Baud rate after adjustment	0x10	First address of ROOM (middle byte)
	18th byte (Note 3)		Baud rate after adjustment	0x00	First address of ROM (lower byte)
	19th byte (Note 3)		Baud rate after adjustment	0x10	End address of ROM (upper byte))
	20th byte (Note 3)		Baud rate after adjustment	0xFF	End address of ROM (middle byte)
	21th byte (Note 3)		Baud rate after adjustment	0xFF	End address of ROM (lower byte)
	22nd byte (Note 3)		Baud rate after adjustment	0x00	First address of RAM (upper byte)
	23rd byte (Note 4)		Baud rate after adjustment	0x00	First address of RAM (middle byte)
	24th byte (Note 4)		Baud rate after adjustment	0x60	First address of RAM (lower byte)
	25th byte (Note 4)		Baud rate after adjustment	0x00	End address of RAM (upper byte)
	26th byte (Note 4)		Baud rate after adjustment	0x0C	End address of RAM (middle byte)
	27th byte (Note 4)		Baud rate after adjustment	0x3F	End address of RAM (lower byte)
	28th byte		Baud rate after adjustment	Checksum of transfer data (complement of 2 of the sum total from 9th through 27th bytes)	
	29th byte	(Wait for the next operation command data)	Baud rate after adjustment	-	

Note 1: “0xyy × 3” means that the device goes into an idle state after transmitting 3 bytes of 0xyy. For further information, refer to Table 26-18.

Note 2: The ROM size code at the 14th byte is shown in Table 26-13.

Note 3: The 16th through 21st bytes show the range of addresses in flash memory where data can be written.

Note 4: The 22th through 27th bytes show the flash memory area and RAM area that can be used by the RAM loader. Because the range of addresses shown here does not include the work area used by BOOTROM, it is smaller than the size of a RAM built into an actual product.

Downloaded from Arrow.com.

Table 26-13 ROM Size Code (14th Byte)

7	6	5	4	3	2	1	0	
ROMSIZE					"0"	"0"	"0"	TMP89FW24AãKíËíl(1111 1000)

ROMSIZE	Flash memory size	00010 : 4 Kbytes 00100 : 8 Kbytes 01000 : 16 Kbytes 10000 : 32 Kbytes 11000 : 48 Kbytes 11110 : 60 Kbytes 10001 : 96 Kbytes 11111 : 124 Kbytes	Read only

Downloaded from Arrow.com.

26.8.7 Flash memory status output command (0xC3)

Table 26-14 shows the transfer format of the flash memory status output command.

Table 26-14 Flash Memory Status Output Command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller	
BOOT ROM	1st byte 2nd byte	Matching data 1(0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	-(Automatic baud rate adjustment) OK:Echo back data (0x86 or 0x30) Error: No data transmitted	
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted	
	5th byte 6th byte	Operation command data (0xC3) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0xC3) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)	
	7th byte		Baud rate after adjustment	0x3A	Start mark
	8th byte		Baud rate after adjustment	0x04	Byte count (from 9th to 12 bytes)
	9th byte		Baud rate after adjustment	Status code 1	
	10th byte		Baud rate after adjustment	Status code 2	
	11th byte		Baud rate after adjustment	0x00	Reserved
	12th byte		Baud rate after adjustment	0x00	Reserved
	13th byte		Baud rate after adjustment	Checksum (complement of 2 of the sum total from 9th through12th bytes)	
	14th byte	(Wait for the next operation command data)	Baud rate after adjustment	-	

Note 1: "0xyy × 3" means that the device goes into an idle state after transmitting 3 bytes of 0xyy.

Note 2: For detailed information on status codes 1 and 2, refer to "26.8.7.1 Flash memory status code".

Downloaded from Arrow.com.

26.8.7.1 Flash memory status code

The flash memory status code is 7-byte data. If shows the status of Security Program and the status of addresses 0x1FFE0 through 0x1FFFF.

Table 26-15 Flash Memory Status Code

Data	Description	In the case of TMP89FW24A
1st	Start mark	0x3A
2nd	Number of transfer data (4 bytes from 3rd through 6th bytes)	0x04
3rd	Status code 1	See below.
4th	Status code 2	See below.
5th	Reserved	0x00
6th	Reserved	0x00
7th	Checksum of transfer data (complement of 2 of the sum total of 3rd through 6th bytes)	If 3rd data is 0x00, then checksum is 0x00. If 3rd data is 0x01, then checksum is 0xFF. If 3rd data is 0x02, then checksum is 0xFE. If 3rd data is 0x03, then checksum is 0xFD.

Status code 1

7	6	5	4	3	2	1	0	
				EPFC	DAFC	RPENA	BLANK	Initial value (**** ****)

EPFC	Password string evaluation when the flash memory erase command is executed (status of 0x1FFFA)	0: 1:	Do not evaluate password string. (Evaluate PNSA and PCSA only.) Evaluate password string, PNSA and PCSA.
DAFC	Security Program check of the on-chip debugging function (OCD) (status of 0x1FFFB)	0: 1:	Do not check Security Program when OCD is activated. Check Security Program when OCD is activated.
RPENA	Status of Security Program in flash memory	0: 1:	Security Program is disabled Security Program is enabled
BLANK	Status of 0x1FFE0 through 0x1FFFF	0: 1:	Addresses 0x1FFE0 through 0x1FFFF all contain 0xFF. Addresses 0x1FFE0 through 0x1FFFF contain any number of bytes that are not 0xFF.

Status code 2

7	6	5	4	3	2	1	0	
"1"	"1"	"1"	"1"	"1"	SB2	SB1	SB0	Initial value (1111 1***)

SB2	Status of SB2	0: 1:	Security Program is enabled. Security Program is disabled.
SB1	Status of SB1	0: 1:	Security Program is enabled. Security Program is disabled.
SB0	Status of SB0	0: 1:	Security Program is enabled. Security Program is disabled.

Note 1: The value to be read as status code 2 directly reflects the value of product ID (0xFF7F).

Note 2: For the definition of each bit, refer to "26.12.2 Security Program".

Downloaded from Arrow.com.

Restrictions are placed on the execution of some operation commands depending on the contents of the status code 1. Detailed information on this is shown in the table below. When Security Program is enabled, the flash memory write, RAM loader, and Sector Erase commands cannot be executed. Before executing these commands, you must perform Chip Erase.

RPENA	BLANK	EPFC	DAFC	Flash memory write, flash memory read, and RAM loader commands	Flash memory sum output, product ID output, and status output commands	Flash memory erase command		Flash memory security setting command
						Chip Erase	Sector Erase	
0	0	0	0	Yes	Yes	Yes	No	No
1	0	0	0	No	Yes	Yes	No	No
0	1	0	*	Password	Yes	Yes	No	Password
		1	*	Password	Yes	Password		Password
1	1	0	*	No	Yes	Yes	No	Password
		1	*	No	Yes	Password	No	Password

Note: Yes: The command can be executed.

Password: Password authentication is required for executing the command.

No: The command cannot be executed.

(After returning echo back to the command, the TMP89FW24A stops communication and goes into an idle state.)

Downloaded from Arrow.com.

26.8.8 Flash memory security setting command (0xFA)

Table 26-16 shows the flash memory security command.

Table 26-16 Flash memory security setting command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 (0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK:Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte 6th byte	Operation command data (0xFA) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0xFA) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)
	7th byte 8th byte	Password count storage address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	9th byte 10th byte	Password count storage address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	11th byte 12th byte	Password count storage address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	Password comparison start address 23 to 16	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	15th byte 16th byte	Password comparison start address 15 to 08	Baud rate after adjustment Baud rate after adjustment	- OK:No data transmitted Error: No data transmitted
	17th byte 18th byte	Password comparison start address 07 to 00	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	19th byte : mth byte	Password string -	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	(n-8)th byte	0x07	Baud rate after adjustment	-
	(n-7)th byte	0x00	Baud rate after adjustment	-
	(n-6)th byte	0x80	Baud rate after adjustment	-
	(n-5)th byte	0x00	Baud rate after adjustment	-
	(n-4)th byte	0x00	Baud rate after adjustment	-
	(n-3)th byte	0x00	Baud rate after adjustment	-
	(n-2)th byte	0x80	Baud rate after adjustment	-
	(n-1)th byte	Security setting value	Baud rate after adjustment	-
	nth byte	Checksum (Complement of two of the sum total of (n-7) th to (n-1)th bytes)	Baud rate after adjustment	-
	(n + 1)th byte	-	Baud rate after adjustment	OK: 0xFB (Note 3) Error: No data transmitted
	(n+2)th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: "0xyy × 3" means that the device goes into an idle state after transmitting 3 bytes of 0xyy.
Note 2: For information on the password, refer to "26.12.1 Password".

Downloaded from Arrow.com.

Note 3: If the flash memory security setting command is executed for a blank product or if a password error occurs for a non-blank product, the TMP89FW24A stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 4: If a communication error occurs during the transfer of a password address or password string, the TMP89FW24A stops communication and goes into an idle state. Therefore, if a password error occurs, initialize the TMP89FW24A by using the $\overline{\text{RESET}}$ pin, and restart the serial PROM mode.

Note 5: If Security Program is disabled in the flash memory, it becomes possible to read ROM data freely in parallel PROM mode. Make sure that you enable Security Program in mass production.

Security setting value (nth byte)

7	6	5	4	3	2	1	0
"1"	"1"	"1"	"1"	"1"	SB2	SB1	SB0

SB2	SB2 security control	0: 1:	Enable Security Program Å| (Security Program remains enabled.)
SB1	SB1 security control	0: 1:	Enable Security Program Å| (Security Program remains enabled.)
SB0	SB0 security control	0: 1:	Enable Security Program Å| (Security Program remains enabled.)

Note 1: Bits 7 to 3 must be set to "1".

Note 2: A bit that has been set to "0" (i.e., Security Program is enabled) must not be set to "0" again.

Note 3: Security Program cannot be disabled by writing a "1" to a bit that has been set to "0" (i.e., Security Program is enabled).

Note 4: For the definition of each bit, refer to "26.12.2 Security Program".

Downloaded from Arrow.com.

26.8.9 Clock change command (operation command: 0xA0)

Table 26-17 shows the transfer format of the clock change command.

Table 26-17 Transfer format of the clock change command

	Transfer byte	Transfer data from the external controller to TMP89FW24A	Baud rate	Transfer data from TMP89FW24A to the external controller
BOOT ROM	1st byte 2nd byte	Matching data 1 (0x86 or 0x30) -	Automatic adjustment Baud rate after adjustment	- (Automatic baud rate adjustment) OK:Echo back data (0x86 or 0x30) Error: No data transmitted
	3rd byte 4th byte	Matching data 2 (0x79 or 0xCF) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0x79 or 0xCF) Error: No data transmitted
	5th byte 6th byte	Operation command data (0xA0) -	Baud rate after adjustment Baud rate after adjustment	- OK: Echo back data (0xA0) Error: 0xA1 × 3, 0xA3 × 3, 0x63 × 3 (Note 1)
	7th byte 8th byte	0x55 (internal high-frequency clock) 0xAA (external high-frequency clock)	Baud rate after adjustment Baud rate after adjustment	- - 0x55 : Internal high-frequency clock 0xAA: External high-frequency clock
	9th byte 10th byte	0x80 -	Baud rate after adjustment Baud rate after adjustment	OK: 0x80 Error: No data transmitted
	11th byte 12th byte	0xFF	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	13th byte 14th byte	0xFF	Baud rate after adjustment Baud rate after adjustment	- OK: No data transmitted Error: No data transmitted
	15th byte	-	Baud rate after adjustment	OK: 0x55 Error: No data transmitted
	16th byte	(Wait for the next operation command data)	Baud rate after adjustment	-

Note 1: “0xyy × 3” means that the device goes into an idle state after transmitting 3 bytes of 0xyy.

Note 2: For information on the checksum, refer to "26.10 Checksum (SUM)".

Note 3: The 7th byte must be 0x55 or 0xAA. If the system clock is not to be changed, send the value corresponding to the current clock. The current clock status can be checked with the status code output command.

Downloaded from Arrow.com.

26.9 Error Codes

Table 26-18 shows the error codes that the TMP89FW24A transmits when it detects errors.

Table 26-18 Error Codes

Data transmitted	Meaning of error data
0x63, 0x63, 0x63	Operation command error
0xA1, 0xA1, 0xA1	Framing error in the received data
0xA3, 0xA3, 0xA3	Overrun error in the received data

Note: If a password error occurs, the TMP89FW24A does not transmit an error code.

Downloaded from Arrow.com.

26.10 Checksum (SUM)

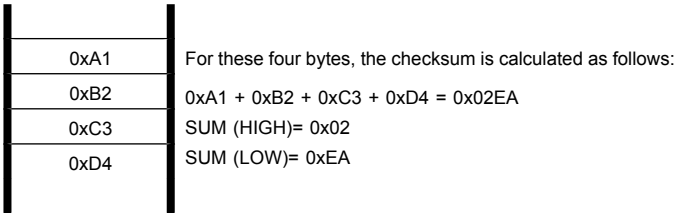
For the following operation commands, a checksum is returned to verify the result of command execution:

- Flash memory erase command (0xF0)
- Flash memory write command (0x30)
- Flash memory SUM output command (0x90)
- Flash memory read command (0x40)
- RAM loader command (0x60)
- Product ID code output command (0xC0)
- Flash memory status output command (0xC3)

26.10.1 Calculation method

The checksum (SUM) is calculated with the sum of all bytes, and the obtained result is returned as a word. In other words, the data is read in byte units and the calculated result is returned in word units.

Example:



In the case of the product ID code output command and flash memory status output command, however, a different calculation method is used. For more information, refer to Table 26-19.

26.10.2 Calculation data

Table 26-19 shows the data to be included in checksum calculation.

Table 26-19 Data To Be Included in Checksum Calculation

Operation command	Calculation data	Note
Flash memory erase command	Data in the entire area of flash memory	
Flash memory write command	Data in the entire area of flash memory	Even if a part of the flash memory is written, the checksum of the entire flash memory area (0x01000 to 0x1FFFF) is calculated. The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Flash memory SUM output command		
Flash memory read command	Data in the read area of flash memory	
RAM loader command	RAM data written in the first received RAM address through the last received RAM address	The data length, address, record type and checksum in Intel Hex format are not included in the checksum.
Product ID code output command	9th through 18th bytes of transferred data	For details, refer to "26.8.6 Product ID code output command (operation command: 0xC0)".
Flash memory status output command	9th through 12th bytes of transferred data	For details, refer to "Table 26-14 Flash Memory Status Output Command".

Downloaded from Arrow.com.

26.11 Intel Hex Format (Binary)

For the following two commands, the Intel Hex format is used in part of the transfer format:

- Flash memory write command (0x30)
- RAM loader command (0x60)

For information on the definition of the Intel Hex format, refer to Table 26-20.

Data is in binary form. The start mark ":" must be transmitted as binary data of 0x3A.

1. After receiving the checksum of each data record, the TMP89FW24A goes into a wait state and awaits the arrival of the start mark ":" (0x3A) of the next data record. If the external controller transmits data other than 0x3A between records, the TMP89FW24A ignores such data when it is in this wait state.

2. If a receiving error or Intel Hex format error occurs, the TMP89FW24A goes into an idle state without returning an error code to the external controller. The Intel Hex format error occurs in the following cases:
   - If the record type is other than 0x00, 0x01 or 0x02
   - If a checksum error of the Intel Hex format occurs
   - If the data length of an extended record (record type = 0x02) is not 0x02
   - If the TMP89FW24A receives the data record after receiving an extended record (record type - 0x02) whose segment address is more than 0x2000
   - If the data length of the end record (record type = 0x01) is not 0x00
   - If the offset address of an extended record (record type = 0x02) is not 0x0000

   The flash memory contained in the TMP89FW24A is comprised of 922 pages, each page containing 128 bytes. Since the flash memory is written page by page (in 128-byte units), if the data to be written to a page is less than 128 Kbytes, the page must be filled up with a given value such as 0xFF. The data to be written to a page can be divided into multiple data records. However, the external controller must observe the following rules when transferring data. Any violation of these rules will cause the device to enter an idle state.
   - The address specified in the first data record after the flash memory write command (0x30) is accepted must be the top address of each page. For example, to write data to Page 1, the address specified in the first data record must be 0x1080 (the top address of Page 1).
   - If the address corresponding to the last byte of the data record is in the middle of a page, the address to be specified in the next data record must be the last address incremented by one.
   - If the address corresponding to the last byte of the data record is the last address of a page, the next record must be an end record or the address to be specified in the next data record must be the top address of a page.
   - The address corresponding to the last byte of the data record immediately before the end record must be the last address of each page.

3. The external controller must be provisioned so that after it transmits the checksum of end record, it goes into a wait state and does not transmit any data until the arrival of 3-byte data (overwrite detection, upper and lower bytes of the checksum) in the case of the flash memory write command. (In the case of the RAM loader command, the external controller awaits the arrival of 2-byte data, upper and lower bytes of the checksum.)

Downloaded from Arrow.com.

Table 26-20 Definition of the Intel Hex Format

	(1)	(2)	(3)	(4)	(5)	(6)
	Start mark	Data length (1 byte)	Offset address (2 bytes)	Record type (1 byte)	Data	Checksum (1 byte)
Data record (record type =00)	3A	Number of bytes in the data field	Starting byte storage address *Specified using big-endian	00	Data (1 to 255 bytes)	Complement of 2 of the sum total of: (2) data length (3) offset address (4) record type (5) data
End record (record type = 01)	3A	00	00 00	01	None	Complement of 2 of the sum total of: (2) data length (3) offset address (4) record type
Extended record (record type =02)	3A	02	00 00	02	Segment address (2 bytes) *Specified using big-endian	Complement of 2 of the sum total of: (3) offset address (4) record type (5) segment address

Downloaded from Arrow.com.

26.12 Security

In serial PROM mode, two security functions are provided to prohibit illegal memory access attempts by a third party: password authentication and Security Program.

26.12.1 Password

Password authentication can be performed in serial PROM mode and when the on-chip debug (OCD) function is used. Arbitrary data in the flash memory (part of user memory) can be specified as a password. When password authentication is enabled in serial PROM mode, the operation commands, such as flash memory write and read, cannot be executed without a valid password. When using the OCD function, password authentication is required.

In parallel PROM mode, password authentication is not supported. To apply access restrictions both in serial PROM and parallel PROM modes, you must also use Security Program.

26.12.1.1 How a password can be specified

The TMP89FW24A allows any piece of data in the flash memory (8 or more consecutive bytes) to be specified as a password. Password authentication is performed by comparing a password string sent from the external controller with the data string in the flash memory specified as a password. Data at addresses 0x01000 through 0x1FEFF in the flash memory can be specified as a password.

26.12.1.2 Password structure

A password consists of three components: PNSA, PCSA, and a password string. Figure 26-4 shows an example of how a password is transmitted.

- PNSA (password count storage address)

  A 3-byte address is specified in the area 0x01000 through 0x1FEFF. The data at the specified address indicates the number of bytes in the password string. If the data is 0x07 or less or the address is outside the designated range, a password error occurs.

  The data at the specified address is defined as N here.

- PCSA (password comparison start address)

  A 3-byte address is specified in the area 0x01000 through (0x1FEFF - N). This address indicates the memory location from where the password string to be compared with a user-entered password. If an address outside the designated range is specified, a password error occurs.

- Password string

  The password string can contain 8 to 255 bytes. A password string sent from the external controller is compared with the password string in the flash memory that is located starting from PCSA and contains the number of bytes (N) specified by PNSA. If the comparison result indicates no match, or if the password string contains three or more consecutive bytes of the same value, a password error occurs and the TMP89FW24A goes into an idle state. In this idle state, external devices cannot communicate with the TMP89FW24A. To resume communication, the TMP89FW24A must be restarted in serial PROM mode by using the reset pin.

Downloaded from Arrow.com.

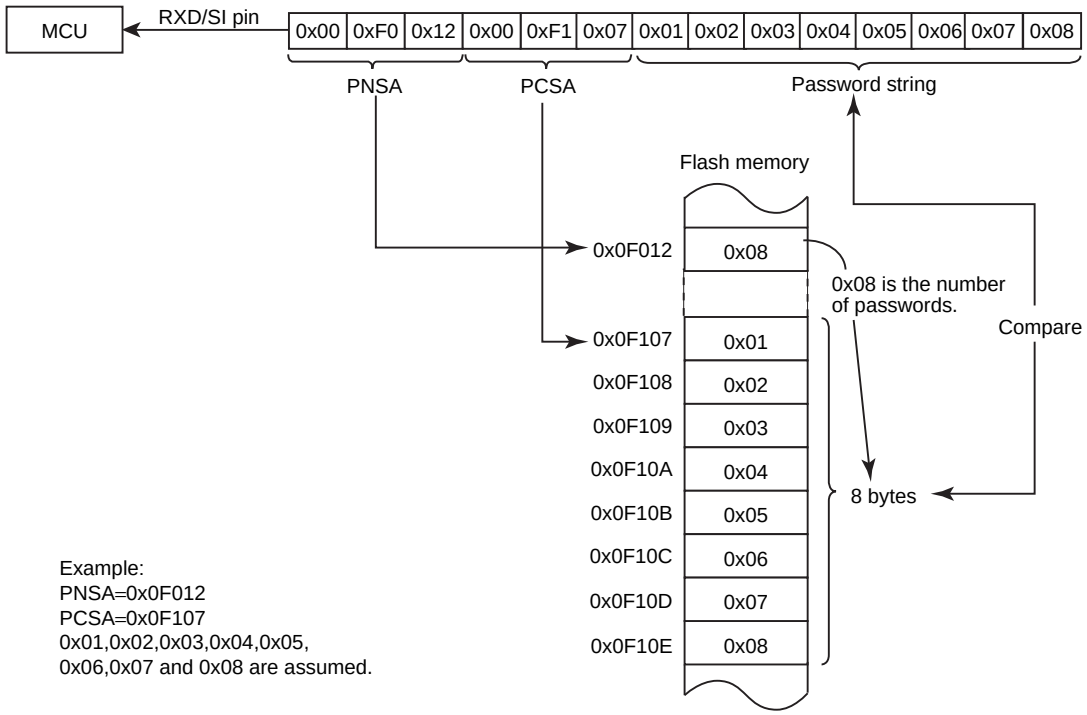


Figure 26-4 Example of How a Password Is Transmitted

Downloaded from Arrow.com.

26.12.1.3 Password setting, clearance and authentication

- Password setting

  Since part of a user program is used as a password, no special processing is required in setting up the device password. The password is automatically set when a user program is written to the flash memory.

- Password clearance

  To clear the password, the entire flash memory must be erased by Chip Erase. When the entire flash memory is initialized to 0xFF, the password is also cleared.

- Password authentication

  If addresses 0x1FFE0 through 0x1FFFF of the TMP89FW24A contain even a single byte of data that is not "0xFF", it is determined to be a non-blank product, and password authentication is required to execute operation commands. Password authentication is performed by using PNSA, PCSA and a password string. Operation commands can be executed only after a valid password is entered. If password authentication fails, the TMP89FW24A goes into an idle state.

  If all the bytes at addresses 0x1FFE0 through 0x1FFFF are "0xFF", the TMP89FW24A is determined to be a blank product, and no password authentication is performed. However, PNSA and PCSA (not a password string) are required to execute particular operation commands. In this case, PNSA and PCSA must be set in the range shown in Table 26-21.

  Whether or not the TMP89FW24A is blank can be checked by executing the status output command.

  The operation commands that require PNSA and PCSA (and a password string) for them to be executed are as follows:

  - Flash memory erase command (0xF0)
  - Flash memory write command (0x30)
  - Flash memory read command (0x40)
  - RAM loader command (0x60)
  - Flash memory security setting command (0xFA)

26.12.1.4 Password values and setting range

A password must be set in accordance with the conditions shown in Table 26-21. If a password created without meeting these conditions is used, a password error occurs. In this cae, the TMP89FW24A does not transmit any data and goes into an idle state.

Table 26-21 Password Values and Setting Range

Password	Blank product (Note 1)	Non-blank product
PNSA (password count storage address)	0x01000 ≤ PNSA ≤ 0x1FEFF	0x01000 ≤ PNSA ≤ 0x1FEFF
PCSA (password comparison start address)	0x01000 ≤ PCSA ≤ 0x1FEFF	0x01000 ≤ PCSA ≤ 0x1FF00 - N
N (password count)	*	8 ≤ N
Password string	Not required (Notes 4 and 5)	Required (Note 3)

Note 1: *: Don’t care.

Note 2: When addresses from 0x1FFE0 through 0x1FFFF are filled with "0sFF", the product is recognized as a blank product.

Downloaded from Arrow.com.

Note 3: A password string must not include three or more consecutive bytes of the same value. (A password error occurs during password authentication. The TMP89FW24A does not transmit any data and goes into a idle state.)

Note 4: When the flash memory write command or RAM loader command is executed on a blank product, the TMP89FW24A receives Intel Hex format data immediately after PCSA without receiving any password string. Even if the external controller sends a dummy password string, the TMP89FW24A ignores received data until the start mark 0x3A (":") of the Intel Hex format arrives. Therefore, the subsequent processing is performed properly. However, if the dummy password string contains "0x3A", it is detected as the start mark erroneously, and the TMP89FW24A goes into an idle state. If this causes any problem, do not transmit a dummy password string.

Note 5: In executing the flash memory erase command, do not transmit a password string to a blank product.

Downloaded from Arrow.com.

26.12.2 Security Program

Security Program can be used in parallel PROM and serial PROM modes and with the OCD function. This protection feature is realized by dedicated memory, and a special command is required to enable it. When SB1 and SB0 of Security Program are enabled, the flash memory cannot be read or written in parallel PROM mode. When SB2 is enabled, the execution of the Page Program command sequence is prohibited in MCU mode. In serial PROM mode, enabling any one of SB2 to SB0 prohibits the execution of the flash memory write, RAM loader, and flash memory read commands. When the OCD function is used, enabling any one of SB2 to SB0 allows you to select whether to prohibit starting up of the system by an option code or to perform password authentication before starting up the system.

26.12.2.1 How Security Program works

The TMP89FW24A allows read protection to be applied to the flash memory by writing protect information to the dedicated memory locations (SB2 to SB0). Since the dedicated memory locations are provided for Security Program, no user resources are required.

26.12.2.2 Enabling or disabling Security Program

- Enabling Security Program

  To enable Security Program, execute the flash memory security setting command.

- Disabling Security Program

  To disable Security Program, execute Chip Erase by the flash memory erase command.

Downloaded from Arrow.com.

26.12.3 Option codes

The following two option codes can be placed at the designated address in the interrupt vector area to enable or disable password authentication for executing the flash memory erase command and to skip Security Program check when the OCD system is started.

- Erase password free code EPFC_OP (0x1FFFA)

  The TMP89FW24A allows password authentication to be performed before the flash memory erase command is executed to prevent illegal memory erasure by third parties. By setting the erase password free code (EPFC_OP), you can enable password authentication for executing the flash memory erase command (0xF0). EPFC_OP is located to address 0x1FFFA in the vector area. To enable password authentication, set EPFC_OP to a value other than 0xFF (typically 0xF0). It is recommended to enable password authentication in mass production.

  In software development, there is a possibility that a password may get lost due to frequent changes to a program. If this happens, you can execute the flash memory erase command (0xF0) without performing password authentication by setting the erase password free code (EPFC_OP) to 0xFF. (Even in this case, PNSA and PCSA must be authenticated.) To disable password authentication for executing the flash memory erase command (0xF0), set EPFC_OP to 0xFF.

  Password authentication can be disabled only for Chip Erase. If Sector Erase is executed with EPFC_OP set to 0xFF, the TMP89FW24A goes into an idle state. Also note that password authentication cannot be disabled for any other operation commands except the flash memory erase command.

- OCD Security Program free code DAFC_OP (0x1FFFB)

  The TMP89FW24A has the Security Program feature to prevent illegal memory accesses by third parties. Security Program, when enabled, blocks any attempts to execute operation commands involving memory access and the startup of the OCD.

  Security Program should normally be enabled at the time of shipment. If you want to use the OCD after shipment while preserving the memory contents, you can use the OCD Security Program free code (DAFC_OP) to start the OCD without checking whether or not Security Program is enabled. (Password authentication is still required even in this case.)

  DAFC_OP is located at address 0x1FFFB in the vector area. To start the OCD without checking the Security Program status, set DAFC_OP to 0xFF. This enables you to start the OCD by performing only password string authentication irrespective of whether or not Security Program is enabled.

  When DAFC_OP is set to a value other than 0xFF, the OCD can only be used when Security Program is disabled. When Security Program is enabled, any attempt to start the OCD causes the TMP89FW24A to stop communication and go into an idle state. To use the OCD while Security Program is enabled, Chip Erase must be performed on the flash memory by executing the flash memory erase command (0xF0).When Security Program is disabled, the OCD can be started by only performing password string authentication.

Table 26-22 Option Codes

Symbol	Function	Address	Set value	
EPFC_OP	Password string authentication when the flash memory erase command is executed	0x1FFFA	0xFF	Skip password string authentication (Authenticate PNSA and PCSA only)
			Other than 0xFF	Authenticate password string, PNSA, and PCSA.
DAFC_OP	Security Program check when the OCD is started	0x1FFFB	0xFF	Skip Security Program check.
			Other than 0xFF	Perform Security Program check.

Downloaded from Arrow.com.

Example :Disabling password authentication for the flash memory erase command and Security Program validation for starting the OCD

Vector Section romdata abs = 0x1FFFA

```
DB      0xFF        ; Disable password authentication for the flash memory erase
                    ; command (EPFC_OP code)
DB      0xFF        ; Skip Security Program check when starting OCD (DAFC_OP code)
```


Downloaded from Arrow.com.

26.12.4 Recommended settings

Table 26-23 and Table 26-24 show the recommended Security Program and option code settings.

Table 26-23 Recommended Settings for Security Program

<table>
<tr><td rowspan="3"></td><td colspan="2">Device status</td><td>MCU mode</td><td>Serial PROM mode</td><td>Parallel PROM mode</td></tr>
<tr><td colspan="2">Security Program</td><td rowspan="2">Page Program command sequence</td><td rowspan="2">Flash memory read, flash memory write and RAM loader commands
(Note 2)</td><td rowspan="2">Page Program command sequence/ flash memory read</td></tr>
<tr><td>SB2</td><td>SB1,SB0
(Note 3)</td></tr>
<tr><td rowspan="2">Software development, debug</td><td colspan="2">Disable</td><td>Can be executed</td><td>Password authentication is required.</td><td rowspan="2">Can be executed
(Note 1)</td></tr>
<tr><td>Enable</td><td>Disable</td><td>Cannot be executed</td><td rowspan="3">Cannot be executed</td></tr>
<tr><td rowspan="2">Mass production</td><td>Disable</td><td>Enable</td><td>Can be executed</td><td rowspan="2">Cannot be executed</td></tr>
<tr><td colspan="2">Enable</td><td>Cannot be executed</td></tr>
</table>

Note 1: When Security Program is disabled (SB1, SB0=0), ROM data can be freely read in parallel PROM mode. In mass production, make sure that you always enable SB1 and SB0. SB1 and SB0 must be enabled.

Note 2: When a program is executed from RAM by using the RAM loader command, the Page Program command sequence cannot be executed as in the case of MCU mode.

Note 3: Security Program is enabled when either or both of SB1 and SB0 are set to "0". Normally, both SB1 and SB0 should be set to "0".

Table 26-24 Recommended Settings for Option Codes

<table>
<tr><td rowspan="3"></td><td colspan="2">Device status</td><td>Serial PROM mode</td><td>Parallel PROM mode</td><td colspan="2">On-chip debug (OCD)</td></tr>
<tr><td colspan="2">Option code</td><td rowspan="2">Flash memory erase command</td><td rowspan="2">Chip Erase command sequence</td><td rowspan="2">SB2 to SB0 are all disabled</td><td rowspan="2">Any one of SB2 to SB0 is enabled</td></tr>
<tr><td>0x1FFFA
(EPFC_OP)</td><td>0x1FFFB
(DAFC_OP)</td></tr>
<tr><td rowspan="2">Software development, debug</td><td rowspan="2">0xFF</td><td>0xFF</td><td rowspan="2">Password string authentication is required.</td><td rowspan="4">Can be executed
(Note 1)</td><td rowspan="4">Can be used</td><td>Can be used</td></tr>
<tr><td>Other than 0xFF</td><td>Cannot be used</td></tr>
<tr><td rowspan="2">Mass production</td><td rowspan="2">Other than 0xFF</td><td>0xFF</td><td rowspan="2">Password string authentication is required.</td><td>Can be used</td></tr>
<tr><td>Other than 0xFF</td><td>Cannot be used</td></tr>
</table>

Note 1: In parallel PROM mode, Chip Erase can be performed irrespective of the option code or Security Program setting.

Downloaded from Arrow.com.

26.13 Flowchart

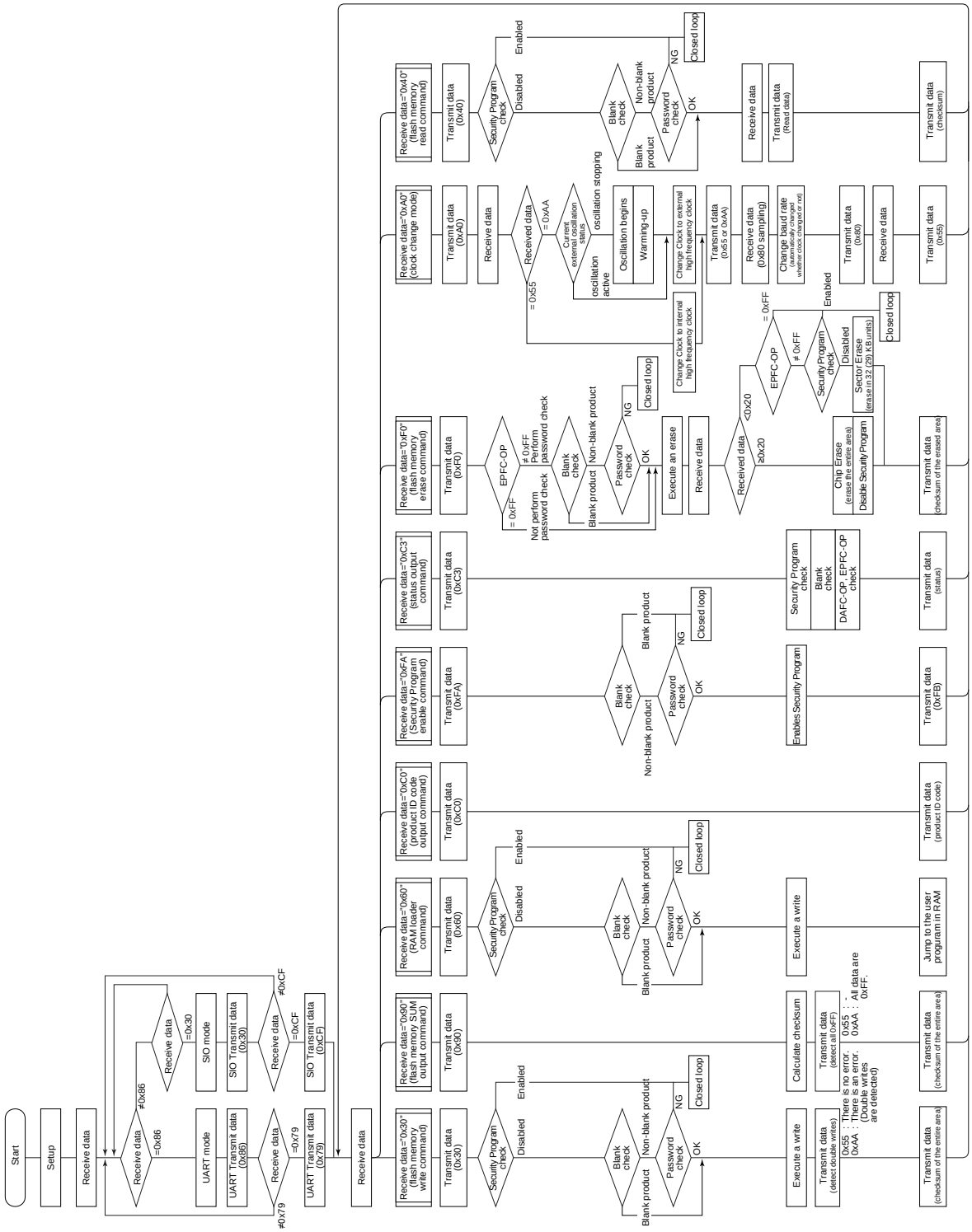


Figure 26-5 Flowchart

Downloaded from Arrow.com.

26.14 AC Characteristics (UART)

Table 26-25 shows the timing from when the MCU receives data from an external device to when the MCU transmits data in return.

Table 26-25 UART Timing -1

Parameter			Symbol	Time (s)		
				Min.	Typ.	Max.
Time from when MCU receives 0x86 to when it echoes back			CMeb1	-	583 / fcgck	-
Time when MCU receives 0x79 to when it echoes back			CMeb2	-	526 / fcgck	-
Time from when MCU receives an operation command to when it echoes back			CMeb3	105 / fcgck	-	241 / fcgck
Time required to calculate the checksum for the flash memory SUM output command			CMfsm	-	5605040 / fcgck	-
Time required to calculate the checksum for the flash memory write command			CMfwr	-	5456240 / fcgck	-
Time required to perform erasure and calculate the checksum for the flash memory erase command	Chip Erase		CMfer	-	(5338530 / fcgck) + 500ms	-
	Sector Erase			-	(5338365 / fcgck) + 100ms	-
Time required to calculate the checksum for the RAM loader command			CMrsm	-	150 / fcgck	-
Time from when MCU receives Intel Hex data to when it sends overwrite detection data			CMwr	182 / fcgck	-	32000 / fcgck + 1.25ms
Time from when MCU receives read byte count data to when it sends memory data			CMrd	-	382 / fcgck	-
Time required to set security by the flash memory security setting command (i =number of bits to be set)			CMrp	-	(857 / fcgck) + (193 x i) / fcgck + (1.25ms x i)	-
Time from when MCU receives clock setting data to when it echoes back	Internal high-frequency → External high-frequency	External high-frequency oscillator stopped	CMck1	-	80531 / fcgck (Note 3)	-
		External high-frequency oscillator active		-	93 / fcgck	-
	External high-frequency → Internal high-frequency	Internal high-frequency oscillator active		-	109 / fcgck	-
Time from when MCU receives 0x80 to when it echoes back			CMck3	-	570 / fcgck	-
Time from when MCU receives 0xFF to when it echoes back			CMck5	-	92 / fcgck	-

Note 1: The security setting time (CMrp) varies with the number of bits to be programmed (i = 1 to 3).

Note 2: The above values indicate the software processing time of the BOOTROM and may include some errors.

Note 3: The above values include the warm-up time until oscillation stabilizes and may vary greatly depending on the characteristics of the oscillator to be used.

Downloaded from Arrow.com.

Table 26-26 shows the time required for the MCU to the acceptance of data.

Table 26-26 UART Timing -2

Parameter	Symbol	Time (s)		
		Min.	Typ.	Max.
Time to hold MODE and RESET pins low after power-on	RSsup	10 ms	-	-
Time from when MODE and RESET pins go high to the acceptance of RXD	RXsup	20 ms	-	-
Time from when MCU echoes back 0x86 to the acceptance of RXD	CMtr1	1 / baud	-	-
Time from when MCU echoes back 0x79 to the acceptance of RXD	CMtr2	(49 / fcgck) + (1 / baud)	-	-
Time from when MCU echoes back an operation command to the acceptance of RXD	CMtr3	1 / baud	-	-
Time from when the execution of a current command is completed to the acceptance of the next operation command	CMnx	(751 / fcgck) + (1 / baud)	-	-
Time from when MCU echoes back clock setting data to the acceptance of RXD	CMck2	(181 / fcgck) + (1 / baud)	-	-
Time from when MCU transmits 0x80 to the acceptance of RXD	CMck4	(35 / fcgck) + (1 / baud)	-	-

Note 1: "1 / baud" indicates the per-bit time of the communication baud rate.

Note 2: When data is transmitted from an external controller to the MCU, make sure a sufficient margin is secured for each of the timing parameters shown above.

Downloaded from Arrow.com.

26.14.1 Reset timing

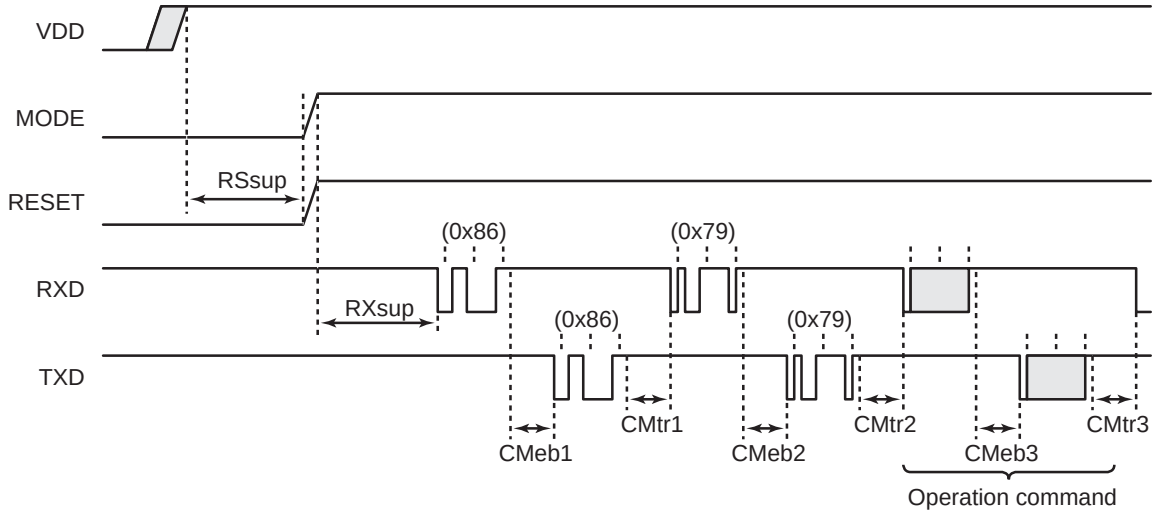


Figure 26-6 Reset Timing

26.14.2 Flash memory erase command (0xF0)

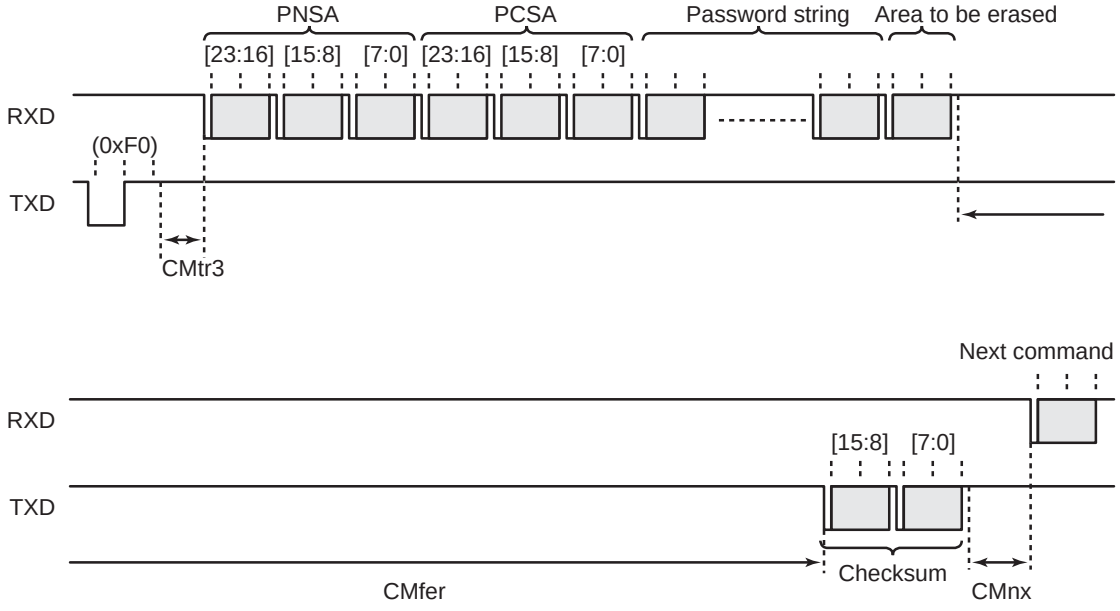


Figure 26-7 Flash Memory Erase Command

Downloaded from Arrow.com.

26.14.3 Flash memory write command (0x30)

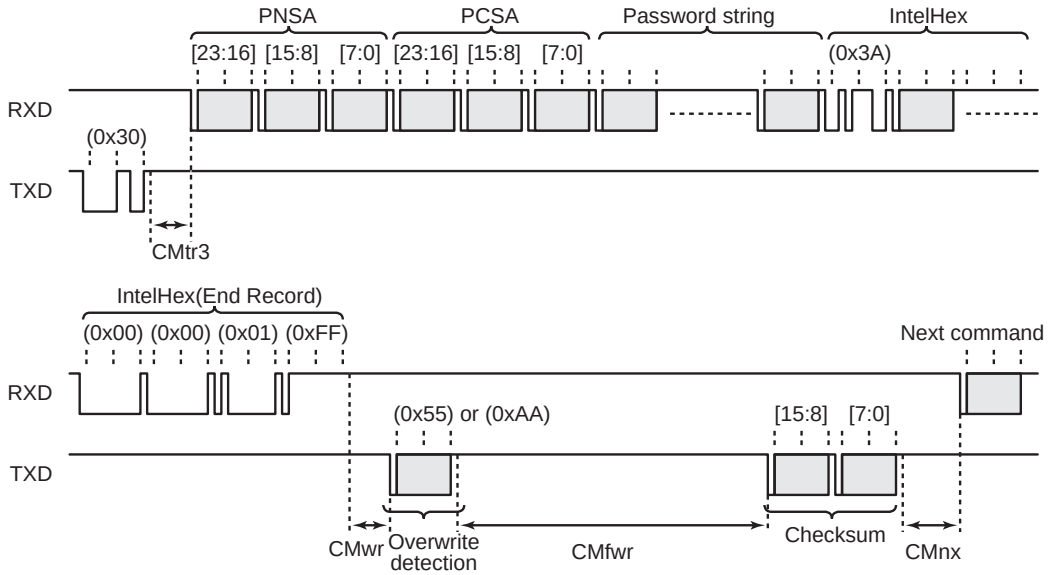


Figure 26-8 Flash Memory Write Command

26.14.4 Flash memory read command (0x40)

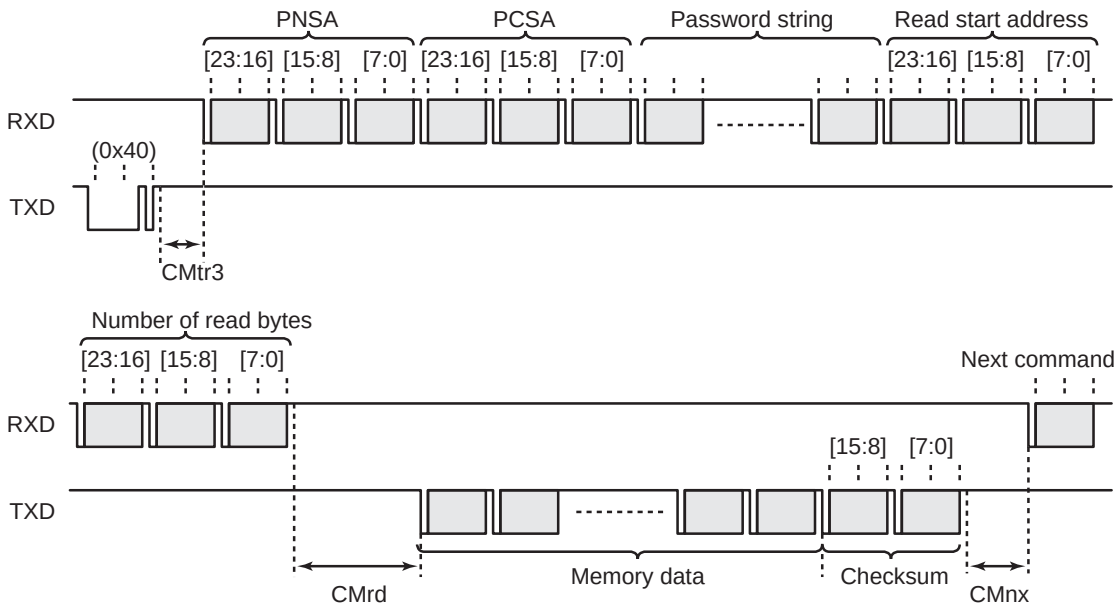


Figure 26-9 Flash Memory Read Command

Downloaded from Arrow.com.

26.14.5 RAM loader command (0x60)

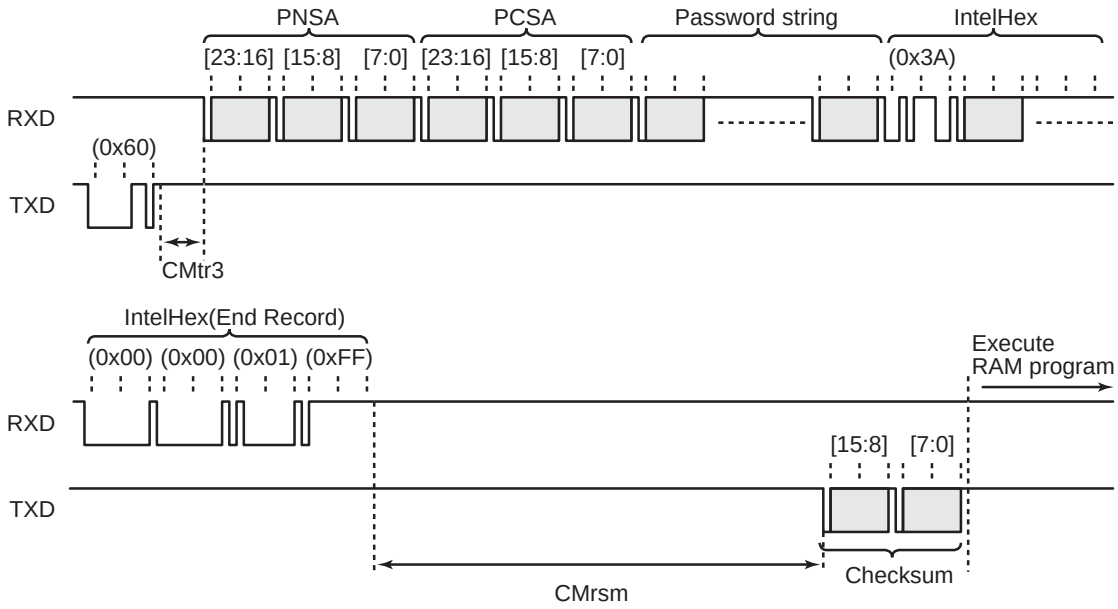


Figure 26-10 RAM Loader Command

26.14.6 Flash memory SUM output command (0x90)

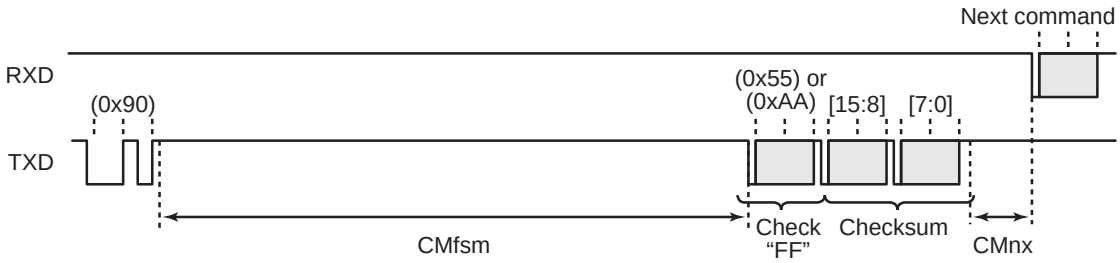


Figure 26-11 Flash Memory SUM Output Command

26.14.7 Product ID code output command (0xC0)

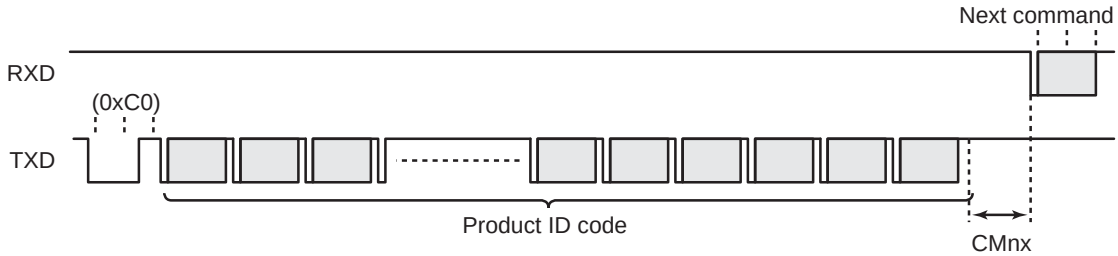


Figure 26-12 Product ID Code Output Command

Downloaded from Arrow.com.

26.14.8 Flash memory status output command (0xC3)

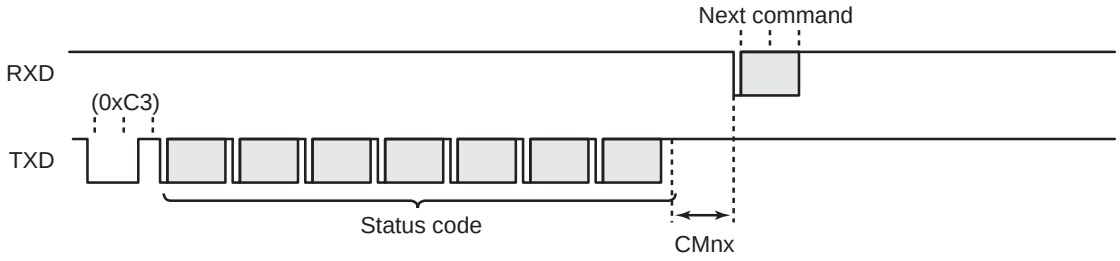


Figure 26-13 Flash Memory Status Output Command

26.14.9 Flash memory security setting command (0xFA)

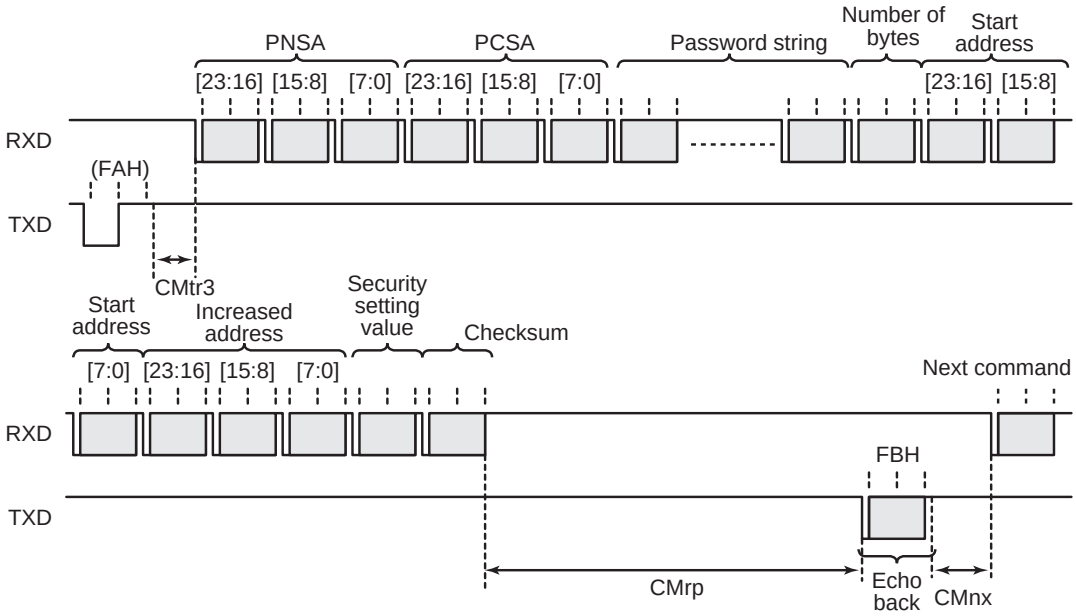


Figure 26-14 Flash Memory Security Setting Command

26.14.10 Clock change command (0xA0)

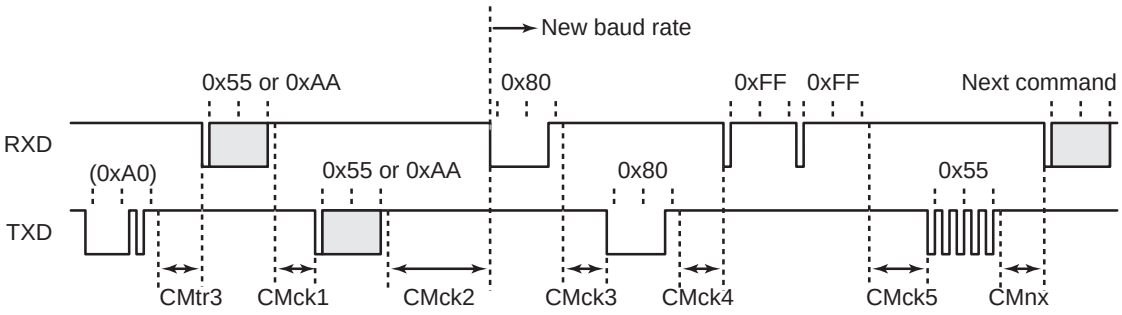


Figure 26-15 Clock Change Command

Downloaded from Arrow.com.

26.15 AC Characteristics (SIO)

Table 26-27 SIO Timing -1

Parameter		Symbol	Time (s)		
			Min.	Typ.	Max.
Time from when MCU transmits 0xCF to when SO0 goes low (busy)		CStr2	-	15 / fcgck	-
Time from when MCU echoes back an operation command to when SC0 goes low (busy)		CStr3	-	15 / fcgck	-
Time required to calculate the checksum for the flash memory SUM output command		CSfsm	-	5609760 / fcgck	-
Time required to calculate the checksum for the flash memory write command		CSfwr	-	5718880 / fcgck	-
Time required to perform erasure and calculate the checksum for the flash memory erase command	Chip Erase	CSfer	-	5365232 / fcgck + 1501 / fcgck + 400ms + 100ms	-
	Sector Erase		-	5616208 / fcgck + 1362 / fcgck + 100ms	-
Time required to set security by the flash memory security setting command (i = number of bits to be set)		CSfrp	-	801 / fcgck + (187 x i) / fcgck + (100ms x i)	-

Note 1: The security setting time (CSfsc) varies with the number of bits to be programmed (i = 1 to 3).

Note 2: The above values include the warm-up time until oscillation stabilizes and may vary greatly depending on the characteristics of the oscillator to be used.

Downloaded from Arrow.com.

Table 26-28 SIO Timing -2

Parameter	Symbol	Time (s)		
		Min.	Typ.	Max.
Time to hold MODE and RESET pins low after power-on	RSsup	-	10 ms	-
Time from when SI0 pin goes high to the acceptance of SI0	RXsup	-	20 ms	-
Time from when MCU completes receiving 0x30 to when SO0 goes low (busy)	CSeb1	-	81 / fcgck	-
Time from when MCU completes receiving 0xCF to when SO0 goes low (busy)	CSeb2	-	44 / fcgck	-
Time from when MCU completes receiving an operation command to when SC0 goes low (busy)	CSeb3	-	41 / fcgck	-
Time from when MCU starts receiving erase range data of the flash memory erase command to when SC0 goes low (busy)	CEeb1	-	107 / fcgck	-
Time from when MCU completes receiving Intel-Hex data of the flash memory write command to when SC0 goes low (busy)	CWeb1	-	79 / fcgck	-
Time from when MCU completes receiving the checksum of the security setting command to when SC0 goes low (busy)	CPeb1	-	63 / fcgck	-
Time from when MCU completes receiving Intel-Hex data of the RAM loader command to when SC0 goes low (busy)	CLeb1	-	65 / fcgck	-
Time from when MCU starts receiving the lower 8 bits of the read byte count of the flash memory read command to when SC0 goes low (busy)	CReb1	-	75 / fcgck	-

Table 26-29 SIO Timing -3

Parameter	Symbol	Time (s)		
		Min.	Typ.	Max.
Transfer clock width (1 bit)	CSbit	40 / fcgck	-	-
Transfer cock width (8 bits)	CSbyte	320 / fcgck	-	-

Downloaded from Arrow.com.

26.15.1 SIO transfer timing

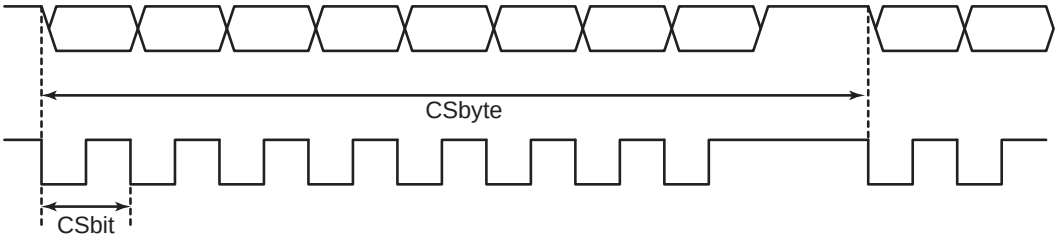


Figure 26-16 SIO transfer timing

26.15.2 Reset timing

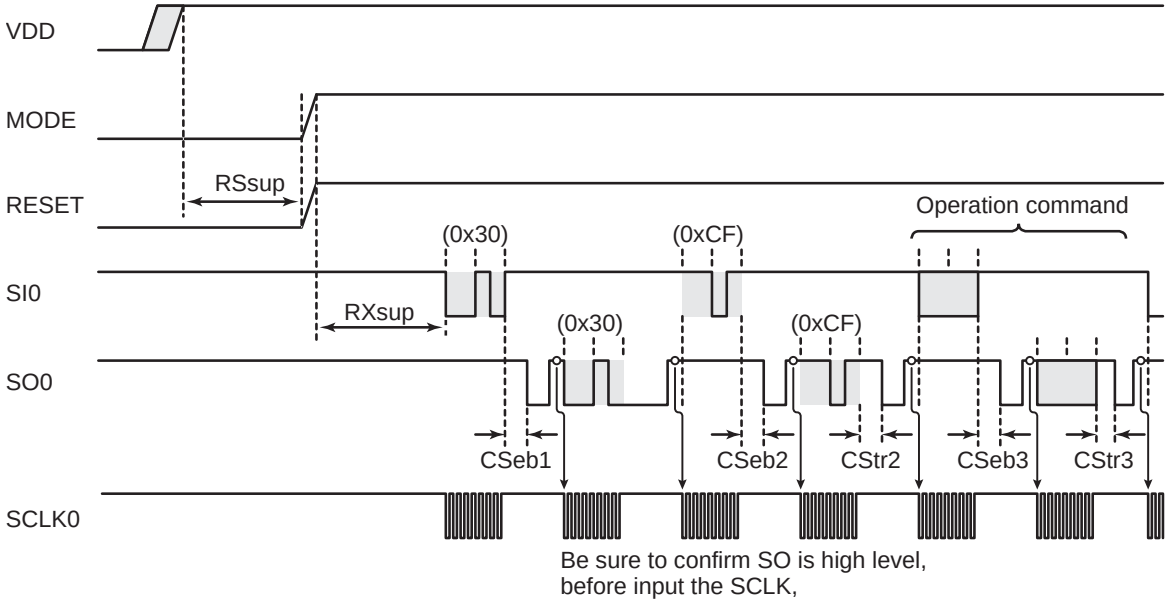


Figure 26-17 Reset Timing

Downloaded from Arrow.com.

26.15.3 Flash memory erase command (0xF0)

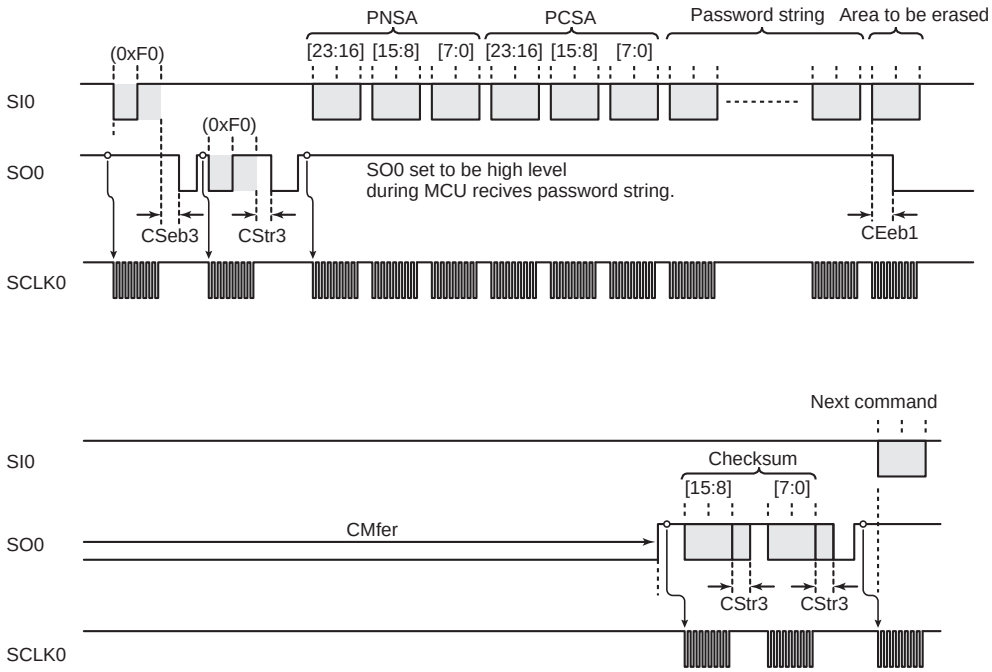


Figure 26-18 Flash Memory Erase Command

Downloaded from Arrow.com.

26.15.4 Flash memory write command (0x30)

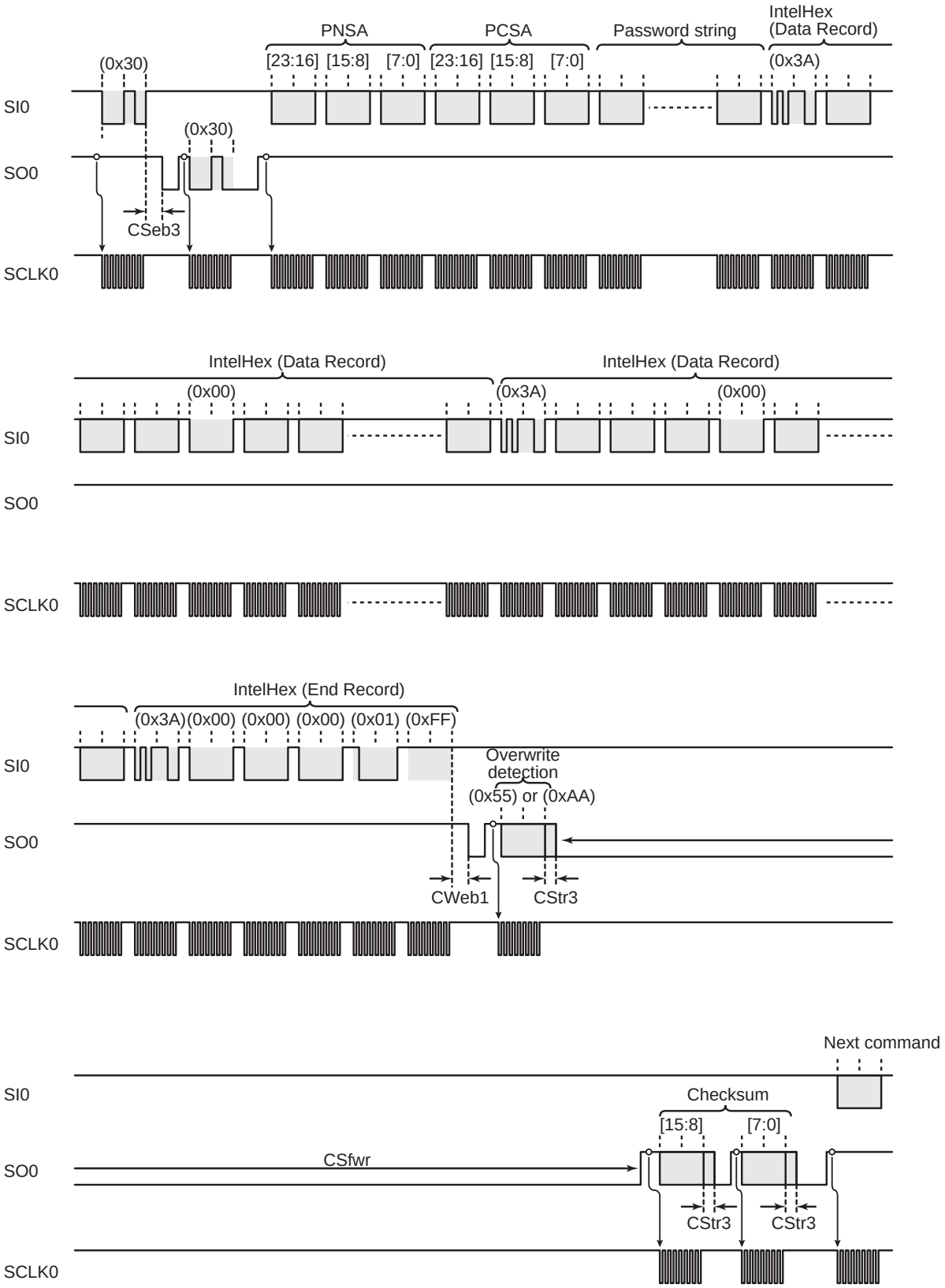


Figure 26-19 Flash Memory Write Command

Downloaded from Arrow.com.

26.15.5 Flash memory read command (0x40)

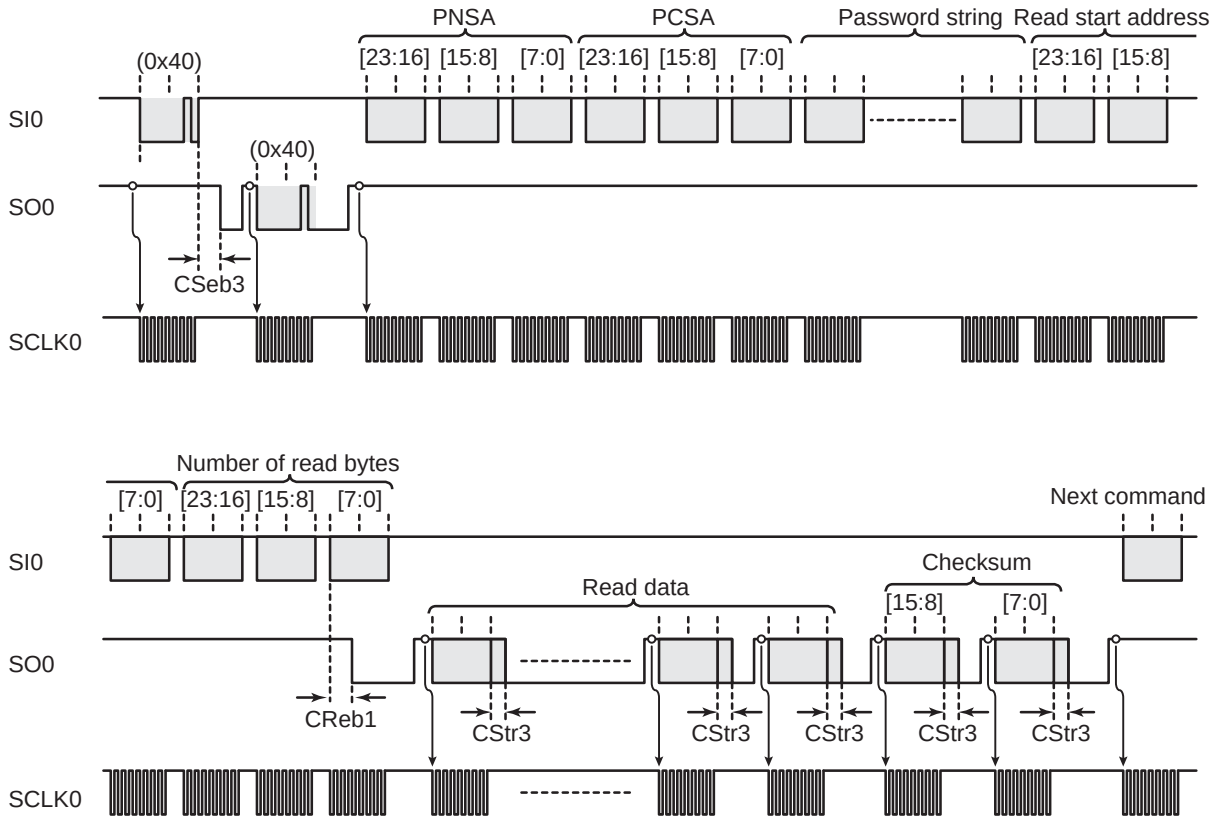


Figure 26-20 Flash Memory Read Command

Downloaded from Arrow.com.

26.15.6 RAM Loader Command (0x60)

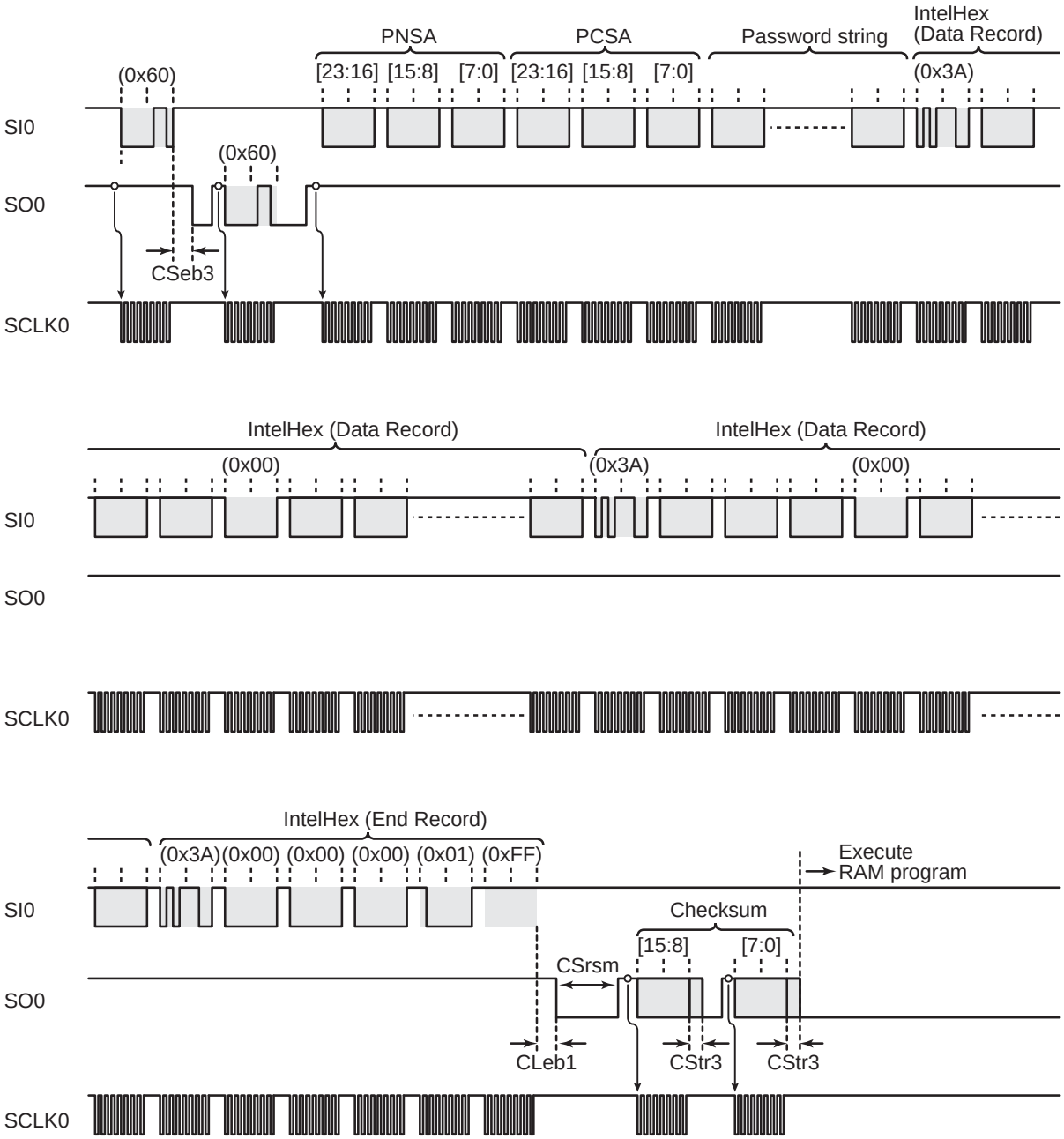


Figure 26-21 RAM Loader Command

Downloaded from Arrow.com.

26.15.7 Flash memory SUM output command (0x90)

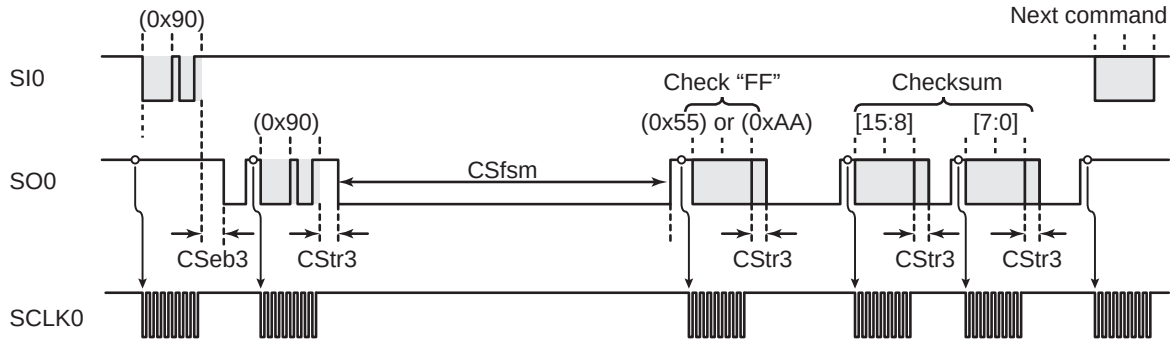


Figure 26-22 Flash Memory SUM Output Command

26.15.8 Product ID code output command (0xC0)

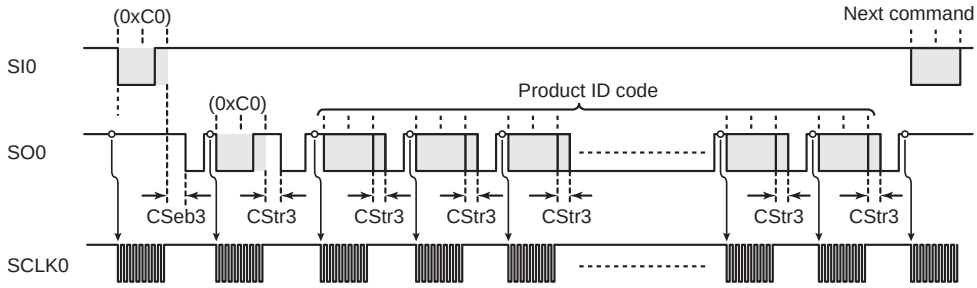


Figure 26-23 Product ID Code Output Command

26.15.9 Flash memory status output command (0xC3)

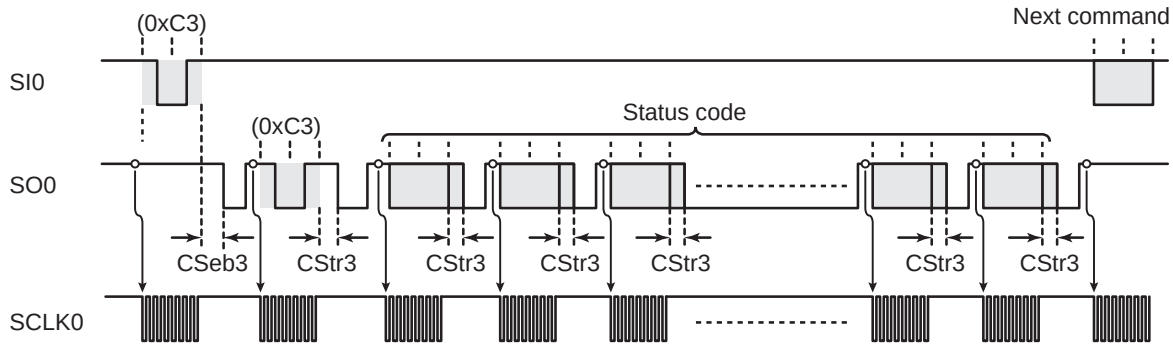


Figure 26-24 Flash Memory Status Output Command

Downloaded from Arrow.com.

26.15.10 Flash memory security setting command (0xFA)

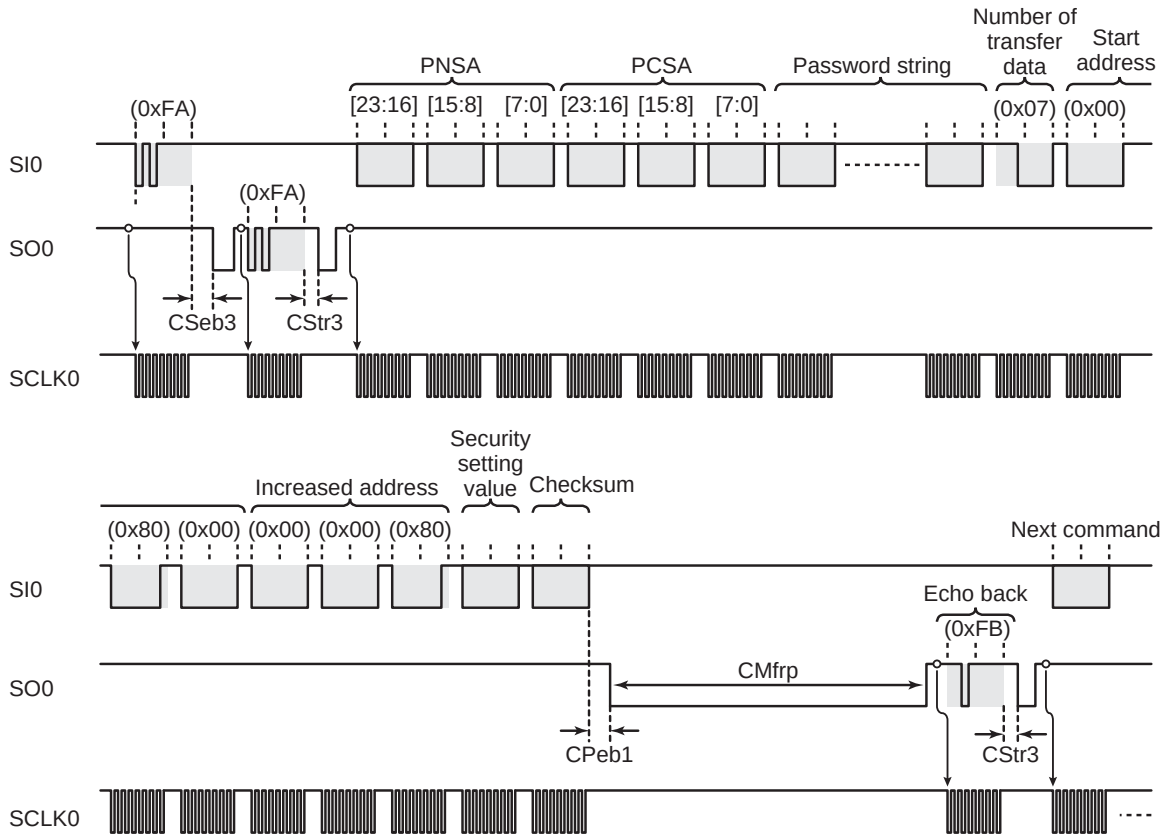


Figure 26-25 Flash Memory Security Setting Command

Downloaded from Arrow.com.

27. On-chip Debug Function (OCD)

The TMP89FW24A has an on-chip debug function. Using a combination of this function and the TOSHIBA on-chip debug emulator RTE870/C1, the user is able to perform software debugging in the on-board environment. This emulator can be operated from a debugger installed on a PC so that the emulation and debugging functions of an application program can be used to modify a program or for other purposes.

This chapter describes the control pins needed to use the on-chip debug function and how a target system is connected to the on-chip debug function. For more detailed information on how to use the on-chip debug emulator RTE870/C1, refer to the emulator operating manual.

27.1 Features

The on-chip debug function of the TMP89FW24A has the following features:

- Debugging can be performed in much the same way as when a microcontroller packaged with the MCU is used.
- The debugging function can be realized using two communication control pins.
- Useful on-chip debug functions include the following:
  - 8 breaks function are provided (one of which can also be used as an event function).
  - A trace function that allows the newest two branch instructions to be stored in real time is provided.
  - Functions to display active memory and to overwrite active memory are provided.
- Built-in flash memory can be erased and written.

27.2 Control Pins

The on-chip debug function uses two pins for communication and four pins for power supply, reset and mode control. The pins used for the on-chip debug function are shown in Table 27-1.

Ports P20 and P21 are used as communication control pins of the on-chip debug function. If the RTE870/C1 On-Chip Debug Emulator is used, therefore, Ports P20 and P21 cannot be debugged as port pins or LCD segment output, UART0 and SIO0 pins. However, because the UART0 and SIO0 functions can be assigned to other ports by using SERSEL<SRSEL2>, these communication functions can also be used during on-chip debug operation. For details, refer to the section of I/O ports.

Table 27-1 Pins Used for the On-chip Debug Function

Pin name (during on-chip debugging)	Input/output	Function		Pin name (in MCU mode)
OCDCK	Input	Communication control pin (clock control)	(Note 1)	P20 / TXD0 / SO0 / SEG37
OCDIO	I/O	Communication control pin (data control)		P21 / RXD0 / SI0 / SEG36
$\overline{\text{RESET}}$	Input	Reset control pin		$\overline{\text{RESET}}$
MODE	Input	Mode control pin		MODE
VDD	Power supply	2.7 V to 5.5 V (Note 1)		
VSS	Power supply	0 V		
Input and output ports other than P20 and P21	I/O	Can be used for an application in a target system		
XIN	Input	To be connected to an oscillator to put these pins in a state of self-oscillation		
XOUT	Output			

Downloaded from Arrow.com.

Note 1: To use all on-chip debug functions, the power supply voltage must be within the range 2.7 V to 5.5 V. If it is within the range 2.2 V to 2.7 V, functional limitations occur with some of the debug functions. For more detailed information, refer to the emulator operating manual.

Downloaded from Arrow.com.

27.3 How to Connect the On-chip Debug Emulator to a Target System

To use the on-chip debug function, the specific pins on a target system must be connected to an external debugging system.

The on-chip debug emulator RTE870/C1 can be connected to a target system via an interface control cable. TOSHIBA provides a connector for this interface control cable as an accessory tool. Mounting this connector on a target system will make it easier to use the on-chip debug function.

The connection between the on-chip debug emulator RTE870/C1 and a target system is shown in Figure 27-1.

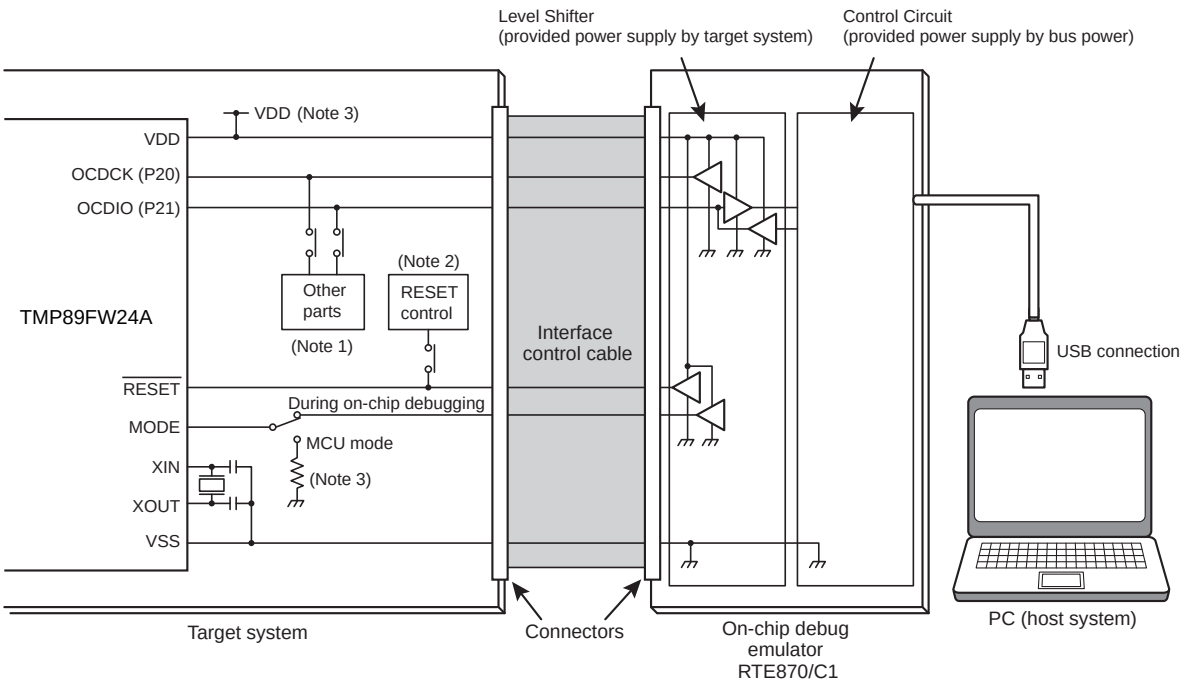


Figure 27-1 How the On-chip Debug Emulator RTE870/C1 Is Connected to a Target System

Note 1: Ports P20 and P21 are used as communication control pins of the on-chip debug function. If the on-chip debug emulator RTE870/C1 is used, therefore, the port functions and the functions of UART0 and SIO0, which are also used as ports, cannot be debugged. If the emulator is disconnected to be used as a single MCU, the functions of ports P20 and P21 can be used. To use the on-chip debug function, however, P20 and P21 should be disconnected using a jumper, switch, etc. if there is the possibility of other parts affecting the communication control.

Note 2: If the reset control circuit on an application board affects the control of the on-chip debug function, it must be disconnected using a jumper, switch, etc.

Note 3: The power supply voltage VDD must be provided by a target system. The VDD pin is connected to the emulator so that the level of voltage appropriate for driving communication pins can be obtained by using the power supply of a target system. The connection of the VDD pin is for receiving the power supply voltage, not for supplying it from the emulator side to a target system.

27.4 Security

The TMP89FW24A provides two security functions to prevent the on-chip debug function from being used through illegal memory access attempted by a third person: a password function and a Security Program function. If a password is set on the TMP89FW24A, it is necessary to authenticate the password for using the on-chip debug function. By setting both a password and the Security Program on the TMP89FW24A, it is possible to prohibit the use of all on-chip debug functions. Furthermore, by using the option code, the on-chip debug function only can be used even if the Security Program is enabled. However, to use the on-chip debug function in this setting, a password authentication process is required.

For information on how to set a password and to enable the read protection and option code, refer to "Serial PROM Mode".

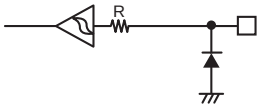
Downloaded from Arrow.com.

Downloaded from Arrow.com.

28. Input/Output Circuit

28.1 Control Pins

The input/output circuitries of the TMP89FW24A control pins are shown below.

Control pin	I/O	Circuitry	Remarks
XIN XOUT	Input Output	Refer to the P0 ports in the chapter of Input/Output Ports.	
XTIN XTOUT	Input Output	Refer to the P0 ports in the chapter of Input/Output Ports.	
$\overline{\text{RESET}}$	Input	Refer to the P1 ports in the chapter of Input/Output Ports.	
MODE	Input		R = 100 Ω (typ.) Hysteresis input

Downloaded from Arrow.com.

Downloaded from Arrow.com.

29. Electrical Characteristics

29.1 Absolute Maximum Ratings

The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

(VSS = 0 V)

Parameter		Symbol	Pins	Ratings	Unit
Supply voltage		V_{DD}		−0.3 to 6.0	V
Input voltage		V_{IN1}	P0, P1, P2, P3, P4, P5, P6, P7, P8, P9 (Three state port)	−0.3 to V_{DD} + 0.3	V
		V_{IN2}	AIN0 to AIN7 (analog input voltage)	−0.3 to A_{VDD}+ 0.3	
Output voltage		V_{OUT1}		−0.3 to V_{DD} + 0.3	V
Output current (per pin)		I_{OUT1}	P0, P1, P2, P3, P4, P5 (Include three state port and pull-up resistor)	-10	mA
		I_{OUT2}	P6, P7, P8, P9 (Include three state port and pull-up resistor)	-10	
		I_{OUT3}	P0, P1, P2, P3, P4, P5 (Include three state port and pull-down resistor)	20	
		I_{OUT4}	P6, P7, P8, P9 (Include three state port and pull-down resistor)	20	
Output current (total)		ΣI_{OUT1}	P0, P1, P2, P3, P4, P5 (Include three state port and pull-up resistor)	-25	
		ΣI_{OUT2}	P6, P7, P8, P9 (Include three state port and pull-up resistor)	-25	
		ΣI_{OUT3}	P0, P1, P2, P3, P4, P5 (Include three state port, pull-down resistor)	40	
		ΣI_{OUT4}	P6, P7, P8, P9 (Include three state port and pull-down)	40	
Power dissipation (Topr = 85 °C)		P_D		250	mW
Soldering temperature (time)		Tsld		260 (10 s)	°C
Storage temperature		Tstg		−55 to 125	
Operating temperature	Except Flash Programming or erasing	Topr		−40 to 85	
	Flash Programming or erasing Serial PROM mode			0 to 70	

Downloaded from Arrow.com.

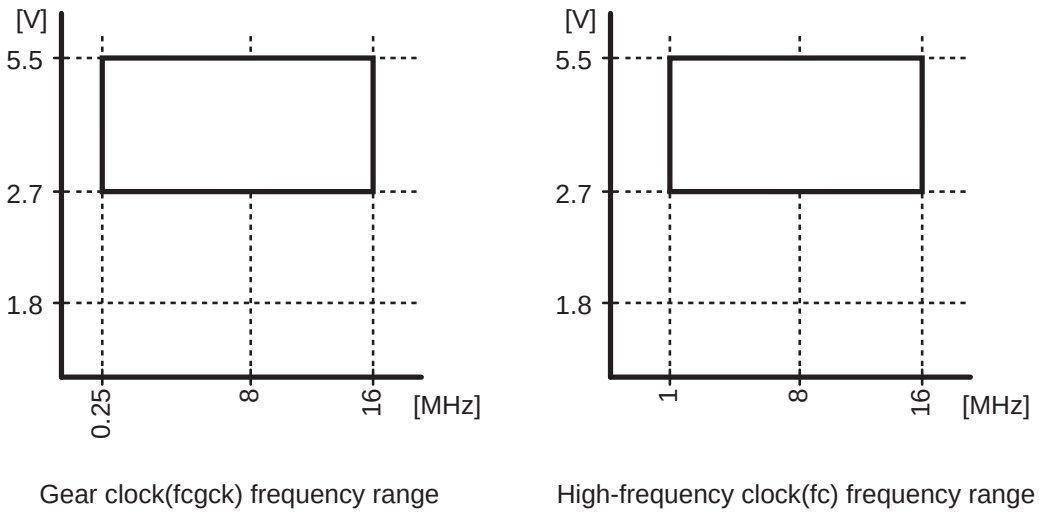
29.2 Operating Conditions

The operating conditions for a device are operating conditions under which it can be guaranteed that the device will operate as specified. If the device is used under operating conditions other than the operating conditions (supply voltage, operating temperature range, specified AC/DC values etc.), malfunction may occur. Thus, when designing products which include this device, ensure that the operating conditions for the device are always adhered to.

29.2.1 MCU mode (Flash Programming or erasing)

(V_{SS} = 0 V, Topr = 0 to 70°C)

Parameter	Symbol	Pins	Condition	Min	Typ	Max	Unit
Supply voltage	V_{DD}		NORMAL1, 2 modes	2.7	–	5.5	V
Input high level	V_{IH1}	MODE pin	V_{DD} ≥ 4.5 V	V_{DD} × 0.70	–	V_{DD}	
	V_{IH2}	Hysteresis input		V_{DD} × 0.75	–		
	V_{IH3}		V_{DD} < 4.5 V	V_{DD} × 0.90	–		
Input low level	V_{IL1}	MODE pin	V_{DD} ≥ 4.5 V	0	–	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input			–	V_{DD} × 0.25	
	V_{IL3}		V_{DD} < 4.5 V		–	V_{DD} × 0.10	
Clock frequency	fc	XIN, XOUT	V_{DD} ≥ 2.7 V	1.0	–	16.0	MHz
	fosc	Internal high-frequency clock		9.5	10.0	10.5	
	fcgck			0.25	–	16.0	



Gear clock(fcgck) frequency range

High-frequency clock(fc) frequency range

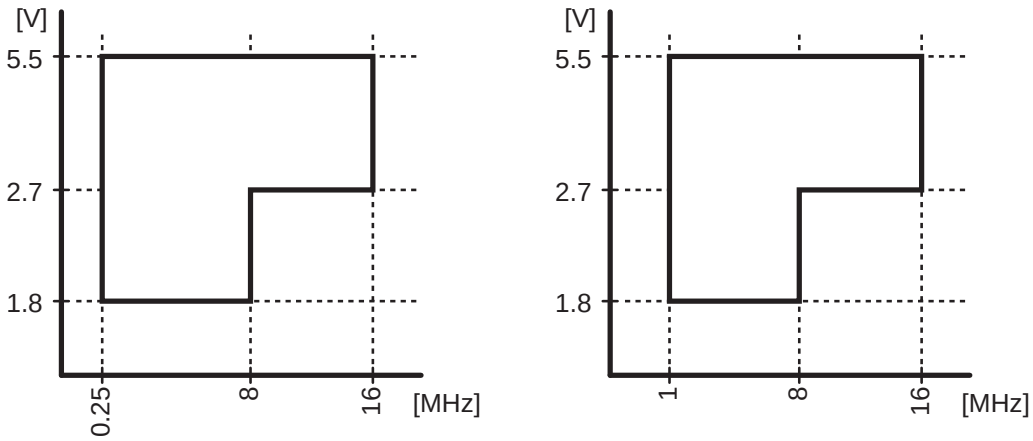
Figure 29-1 Clock gear (fcgck) and External high-frequency clock (fc)

Downloaded from Arrow.com.

29.2.2 MCU mode (Except Flash Programming or erasing)

(V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Pins	Condition		Min	Typ	Max	Unit
Supply voltage	V_{DD}		fc = 16.0 MHz	NORMAL1, 2 modes	2.7	–	5.5	V
			fc = 8.0 MHz	IDLE0, 1, 2 modes	1.8	–		
			fs = 32.768 kHz	SLOW1, 2 modes SLEEP0, 1 modes				
			STOP mode					
Input high leve	V_{IH1}	MODE pin	V_{DD} ≥ 4.5 V		V_{DD} × 0.70	–	V_{DD}	V
	V_{IH2}	Hysteresis input			V_{DD} × 0.75	–		
	V_{IH3}		V_{DD} < 4.5 V		V_{DD} × 0.90	–		
Input low leve	V_{IL1}	MODE pin	V_{DD} ≥ 4.5 V		0	–	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input				–	V_{DD} × 0.25	
	V_{IL3}		V_{DD} < 4.5 V			–	V_{DD} × 0.10	
Clock frequency	fc	XIN, XOUT	V_{DD} = 2.7 to 5.5 V		1.0	–	16.0	MHz
			V_{DD} = 1.8 to 5.5 V			–	8.0	
	fosc	Internal high-frequency clock	V_{DD} = 1.8 to 5.5 V		9.5	10.0	10.5	
	fcgck		V_{DD} = 2.7 to 5.5 V		0.25	–	16.0	
			V_{DD} = 1.8 to 5.5 V			–	8.0	
	fs	XTIN, XTOUT	V_{DD} = 1.8 to 5.5 V		30.0	–	34.0	kHz



Gear clock(fcgck) frequency range

High-frequency clock(fc) frequency range

Figure 29-2 Clock gear (fcgck) and External high-frequency clock (fc)

Downloaded from Arrow.com.

29.2.3 Serial PROM mode

(V_{SS} = 0 V, Topr = 0 to 70°C)

Parameter	Symbol	Pins	Condition	Min	Typ	Max	Unit
Supply voltage	V_{DD}		NORMAL1, 2 modes	2.7	–	5.5	V
Input high level	V_{IH1}	MODE pin	V_{DD} ≥ 4.5 V	V_{DD} × 0.70	–	V_{DD}	
	V_{IH2}	Hysteresis input		V_{DD} × 0.75	–		
	V_{IH3}		V_{DD} < 4.5 V	V_{DD} × 0.90	–		
Input low leve	V_{IL1}	MODE pin	V_{DD} ≥ 4.5 V	0	–	V_{DD} × 0.30	
	V_{IL2}	Hysteresis input			–	V_{DD} × 0.25	
	V_{IL3}		V_{DD} < 4.5 V		–	V_{DD} × 0.10	
Clock frequency	fc	XIN, XOUT	V_{DD} ≥ 2.7 V	1.0	–	16.0	MHz
	fosc	Internal high-frequency clock		9.5	10.0	10.5	
	fcgck			0.25	–	16.0	

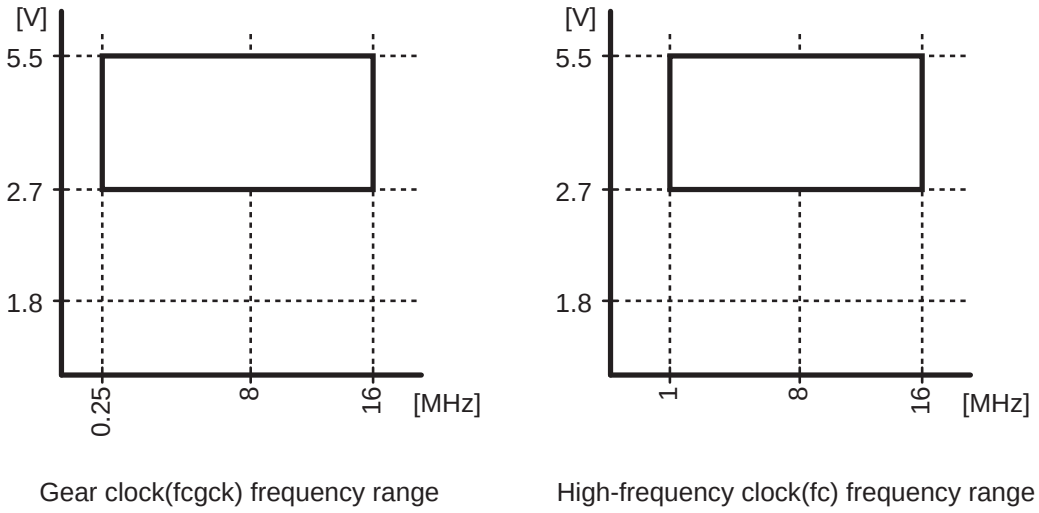


Figure 29-3 Clock gear(fcgck) and External high-frequency clock (fc)

Downloaded from Arrow.com.

29.3 DC Characteristics

(V_{SS} = 0 V, Topr = −40 to 85 °C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Hysteresis voltage	V_{HS}	Hysteresis input	V_{DD} = 5.5 V V_{IN} = 5.5 V/0 V	−	0.9	−	V
Input current	I_{IN1}	MODE		−	−	±2	μA
	I_{IN2}	P0, P1, P2, P4, P5, P7, P8, P9, PB					
	I_{IN3}	$\overline{RESET}$, $\overline{STOP}$					
Input resistance	R_{IN2}	$\overline{RESET}$ pull-up	V_{DD} = 5.5 V, V_{IN} = 0 V	100	220	500	kΩ
	R_{IN3}	P0, P1, P2, P8		30	50	100	
Output leakage current	I_{LO2}	P0, P1, P2, P3, P4, P5, P6, P7, P8, P9	V_{DD} = 5.5 V, V_{OUT} = 5.5 V/0 V	−	−	±2	μA
Output high voltage	V_{OH}	P0, P1, P2, P3, P4, P5, P6, P7, P8, P9 (Except XOUT, XTOUT)	V_{DD} = 4.5 V, I_{OH} = −0.7mA	4.1	−	−	V
Output low voltage	V_{OL}	P0, P1, P2, P3, P4, P5, P6, P7, P8, P9 (Except XOUT, XTOUT)	V_{DD} = 4.5 V, I_{OL} = 1.6 mA	−	−	0.4	V
Output low current	I_{OL}	P9, P84 to P87	V_{DD} = 4.5 V, V_{OL} = 1.0 V	−	10	−	mA

Note 1: Typical values show those at Topr = 25°C and V_{DD} = 5.0 V.

Note 2: Input current I_{IN3} : The current through pull-up resistor is not included.

Downloaded from Arrow.com.

(V_{SS} = 0 V, Topr = −40 to 85 °C)

Parameter	Symbol	Pins	Condition			Min	Typ.	Max	Unit
Supply current in NORMAL 1, 2 modes	I_{DD}		V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2V fcgck = 16.0 MHz fs = 32.768 kHz (When external high frequency clock operation.)	SDWCR1<FLSOFF>="0" (When a program operates on flash memory)		−	4.4	7.0	mA
				SDWCR1<FLSOFF>="1" (When a program operates on RAM)		−	3.5	5.6	
Supply current in IDLE1, 2 modes				SDWCR1<FLSOFF>="0"		−	3.6	6.0	
				SDWCR1<FLSOFF>="1"		−	3.0	5.2	
Supply current in IDLE0 modes				SDWCR1<FLSOFF>="0"		−	2.4	4.6	
				SDWCR1<FLSOFF>="1"		−	1.8	3.9	
Supply current in NORMAL1, 2 modes			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2V fcgck = 10.0 MHz fs = 32.768 kHz (When internal high frequency clock operation.)	SDWCR1<FLSOFF>="0" (When a program operates on flash memory)		−	3.3	3.8	
				SDWCR1<FLSOFF>="1" (When a program operates on RAM)		−	2.5	2.8	
Supply current in IDLE1, 2 modes				SDWCR1<FLSOFF>="0"		−	2.7	3.2	
				SDWCR1<FLSOFF>="1"		−	1.8	2.0	
Supply current in IDLE0 modes				SDWCR1<FLSOFF>="0"		−	1.8	2.3	
				SDWCR1<FLSOFF>="1"		−	1.4	1.6	
Supply current in SLOW1 mode			V_{DD} = 3.0 V V_{IN} = 2.8 V/0.2 V fs = 32.768 kHz	SDWCR1<FLSOFF>="0" (When a program operates on flash memory)		−	550	850	μA
				SDWCR1<FLSOFF>="1" VLTD Enable	Topr=25°C	−	9	12	
					Topr=85°C	−	−	60	
				SDWCR1<FLSOFF>="1" VLTD Disable	Topr=25°C	−	8	11	
					Topr=85°C	−	−	60	
Supply current in SLEEP1 mode				SDWCR1<FLSOFF>="0"		−	490	630	
				SDWCR1<FLSOFF>="1"		−	8	60	
Supply current in SLEEP0 mode				SDWCR1<FLSOFF>="0"		−	490	620	
				SDWCR1<FLSOFF>="1"		−	8	60	
Supply current in STOP mode			V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V		VLTD Enable	−	10	50	
					VLTD Disable	−	8	50	
Current for writing to flash memory, erasing and security program	I_{DDEW}		V_{DD} = 5.5 V V_{IN} = 5.3 V/0.2 V			−	4.2	−	mA

Note 1: Typical values shown are Topr = 25°C and V_{DD} = 5.0 V, unless otherwise specified.

Note 2: I_{DD} does not include I_{REF}. It is the electrical current in the state in which the peripheral circuitry has been operated.

Note 3: Each supply current in SLOW2 mode is equivalent to that in IDLE0, IDLE1 and IDLE2 modes.

Note 4: VLTD (Voltage detection circuit)

Note 5: Writing, erasing and security program temperature range is 0 to 70°C.

Downloaded from Arrow.com.

29.4 AD Conversion Characteristics

(V_{SS} = 0.0 V, 4.5 V ≤ V_{DD} ≤ 5.5 V, Topr = −40 to 85 °C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		AVDD-1.0	−	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
Analog reference voltage range (Note4)	Δ V_{AREF}		3.5	−	−	
Analog input voltage range	V_{AIN}		V_{SS}	−	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 5.5 V V_{SS} = A_{VSS} = 0.0 V	−	0.60	0.76	mA
Non-linearity error		V_{DD} = A_{VDD} = 5.0 V, V_{SS} = A_{VSS} = 0.0 V V_{AREF} = 5.0 V	−	−	±2	LSB
Zero point error			−	−	±2	
Full scale error			−	−	±2	
Total error			−	−	±2	

(V_{SS} = 0.0 V, 2.7 V ≤ V_{DD} < 4.5 V, Topr = −40 to 85 °C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		AVDD-1.0	−	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
Analog reference voltage range (Note4)	Δ V_{AREF}		2.5	−	−	
Analog input voltage range	V_{AIN}		V_{SS}	−	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 4.5 V V_{SS} = A_{VSS} = 0.0 V	−	0.49	0.62	mA
Non-linearity error		V_{DD} = A_{VDD}= 2.7 V V_{SS} = A_{VSS} = 0.0 V V_{AREF} = 2.7 V	−	−	±2	LSB
Zero point error			−	−	±2	
Full scale error			−	−	±2	
Total error			−	−	±2	

(V_{SS} = 0.0 V, 2.2 V ≤ V_{DD} < 2.7 V, Topr = −40 to 85 °C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	V_{AREF}		AVDD-0.5	−	A_{VDD}	V
Power supply voltage of analog control circuit	A_{VDD}		V_{DD}			
Analog reference voltage range (Note4)	Δ V_{AREF}		2.5	−	−	
Analog input voltage range	V_{AIN}		V_{SS}	−	V_{AREF}	
Power supply current of analog reference voltage	I_{REF}	V_{DD} = A_{VDD} = V_{AREF} = 2.7 V V_{SS} = A_{VSS} = 0.0 V	−	0.29	0.37	mA
Non-linearity error		V_{DD} = A_{VDD} = 2.2 V V_{SS} = A_{VSS} = 0.0 V V_{AREF} = 2.2 V	−	−	±4	LSB
Zero point error			−	−	±4	
Full scale error			−	−	±4	
Total error			−	−	±4	

Note 1: The total error includes all errors except a quantization error, and is defined as the maximum deviation from the ideal conversion line.

Note 2: Conversion times differ with variation in the power supply voltage.

Note 3: The voltage to be input to the AIN input pin must be within the range V_{AREF} to V_{SS}. If a voltage outside this range is input, converted values will become indeterminate, and converted values of other channels will be affected.

Note 4: Analog reference voltage range: $\Delta V_{AREF} = V_{AREF} - V_{SS}$

Note 5: If the AD converter is not used, fix the A_{VDD} pins to the V_{DD} level.

Downloaded from Arrow.com.

29.5 Power-on Reset Circuit Characteristics

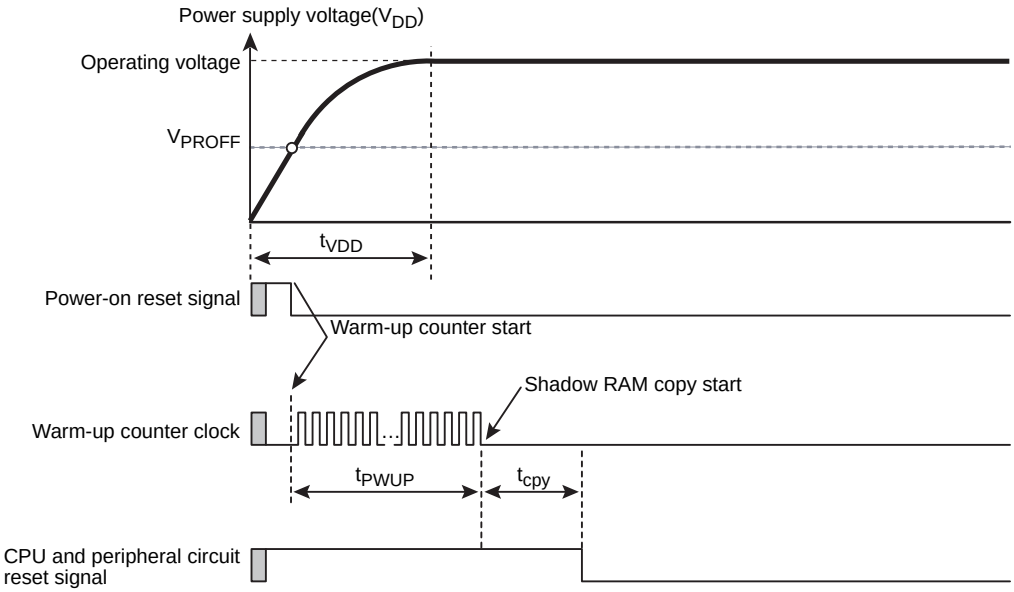


Figure 29-4 Power-on Reset Operation Timing

Note: Care must be taken in system designing since the power-on reset circuit may not fulfill its functions due to the fluctuations in the power supply voltage (V_{DD}).

(V_{SS} = 0 V, Topr = −40 to 85°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{PROFF}	Power-on reset releasing voltage	−	1.5	−	V
t_{PWUP}	Warming-up time after a reset is cleared	−	102 x 2^9/fosc	−	s
t_{cpy}	Shadow RAM copying time	−	3072/(fcgck x 4)	−	s
t_{VDD}	Power supply rise time	−	−	5	ms

Note 1: A clock output by an oscillating circuit is used as the input clock for a warming-up counter. Because the oscillation frequency does not stabilize until an oscillating circuit stabilizes, some errors may be included in the warming-up time.

Note 2: Boost the power supply voltage such that t_{VDD} becomes smaller that t_{PWUP}.

Downloaded from Arrow.com.

29.6 Voltage Detecting Circuit Characteristics

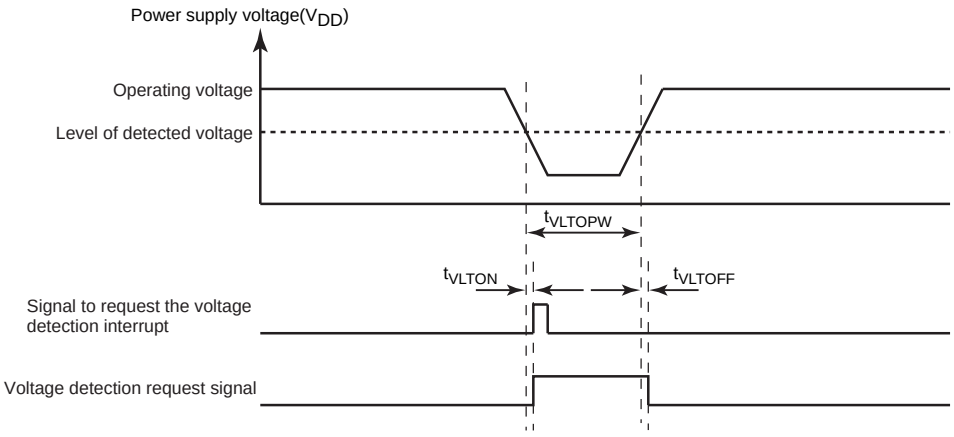


Figure 29-5 Operation Timing of the Voltage Detecting Circuit

Note: Care must be taken in system designing since the power-on reset circuit may not fulfill its functions due to the fluctuations in the power supply voltage (V_{DD}).

(V_{SS} = 0 V, Topr = −40 to 85°C)

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{VLTOFF}	Voltage detection releasing response time	−	150	300	us
t_{VLTON}	Voltage detecting detection response time	−	40	60	
t_{VLTPW}	Voltage detecting minimum pulse width	10	−	−	

Downloaded from Arrow.com.

29.7 16-bit Timer Counter(TCB) Characteristics

(V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition		Min	Typ.	Max	Unit
TCB0 pin input pulse width	t_{TCB0}		NORMAL1,2 modes IDLE1, 2 modes	2 / fcgck	−	−	s
			SLOW1, 2 modes SLEEP1 modes	2 / fs	−	−	

29.8 LCD Characteristics

(V_{SS} = 0 V, Topr = −40 to 85°C)

Parameter	Symbol	Pins	Condition	Min	Typ.	Max	Unit
Supply voltage	V_{DD}			1.8	-	5.5	V
Power supply for LCD driver	V_{LC}	VLC	LCD driver is enable	2.2	-	V_{DD}	V
			LCD driver is not enable	1.8	-	V_{DD}	V
Internal bleeder resistance	R_{H1}			375	500	625	kΩ
	R_{H2}			150	200	250	
	R_L			15	20	25	

Note 1: LCD driver should be disabled when the power supply for LCD driver (V_{LC}) falls to or below the 2.2V.

Note 2: R_{H1},R_{H2}:High resistance

Note 3: R_L:Low resistance

Downloaded from Arrow.com.

29.9 AC Characteristics

29.9.1 MCU mode (Flash programming or erasing)

(V_{SS} = 0 V, V_{DD} = 2.7 V to 5.5 V, Topr = 0 to 70°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.0625	–	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	–	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external high-frequency clock operation (XIN input). fc = 16.0MHz	–	31.25	–	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external low-frequency clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low-level clock pulse width	t_{WSL}					

29.9.2 MCU mode (Except Flash Programming or erasing)

(V_{SS} = 0 V, V_{DD} = 2.7 V to 5.5 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.0625	–	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	–	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external high-frequency clock operation (XIN input). fc = 16.0MHz	–	31.25	–	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external low-frequency clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low-level clock pulse width	t_{WSL}					

(V_{SS} = 0 V, V_{DD} = 1.8 V to 5.5 V, Topr = −40 to 85°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.125	–	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	–	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external high-frequency clock operation (XIN input). fc = 8.0 MHz	–	62.5	–	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external low-frequency clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low-level clock pulse width	t_{WSL}					

Downloaded from Arrow.com.

29.9.3 Serial PROM mode

(V_{SS} = 0 V, V_{DD} = 2.7 V to 5.5 V, Topr = 0 to 70°C)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Machine cycle time	t_{cy}	NORMAL1, 2 modes	0.0625	–	4	μs
		IDLE0, 1, 2 modes				
		SLOW1, 2 modes	117.6	–	133.3	
		SLEEP0, 1 modes				
High-level clock pulse width	t_{WCH}	For external high-frequency clock operation (XIN input). fc = 16.0MHz	–	31.25	–	ns
Low-level clock pulse width	t_{WCL}					
High-level clock pulse width	t_{WSH}	For external low-frequency clock operation (XTIN input) fs = 32.768 kHz	–	15.26	–	μs
Low-level clock pulse width	t_{WSL}					

29.10 Flash Characteristics

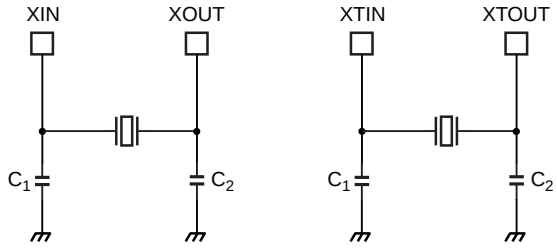
29.10.1 Write characteristics

(V_{SS} = 0 V, Topr = 0 to 70°C)

Parameter	Condition	Min	Typ.	Max.	Unit
Number of guaranteed writes to flash memory		–	–	100	Times
Flash memory write time	Per 1 page (128 bytes)	–	1.25	3	ms
Flash memory erase time	Chip Erase	–	400	1000	
	Sector Erase	–	100	250	

Downloaded from Arrow.com.

29.11 Recommended Oscillating Condition



(1) High-frequency oscillation (2) Low-frequency oscillation

Note 1: To ensure stable oscillation, the resonator position, load capacitance, etc. must be appropriate. Because these factors are greatly affected by board patterns, please be sure to evaluate operation on the board on which the device will actually be mounted.

Note 2: The product numbers and specifications of the resonators supplied by Murata Manufacturing Co., Ltd. are subject to change. For up to date information, please refer to the following http://www.murata.com

Downloaded from Arrow.com.

29.12 Handling Precaution

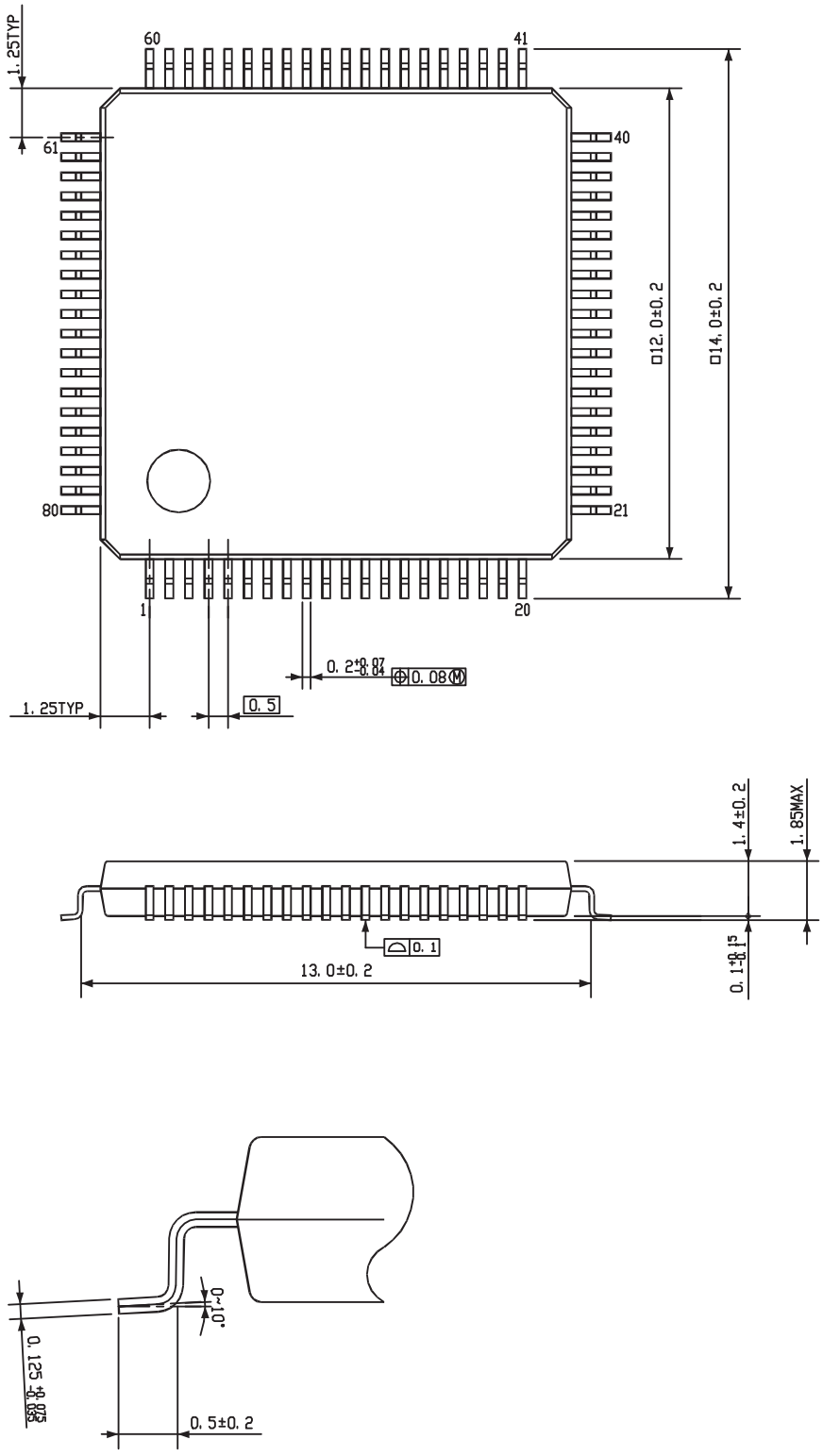
- When using the device (oscillator) in places exposed to high electric fields such as cathode-ray tubes, we recommend electrically shielding the package in order to maintain normal operating condition.

Downloaded from Arrow.com.

30. Package Dimensions

LQFP80-P-1212-0.50F

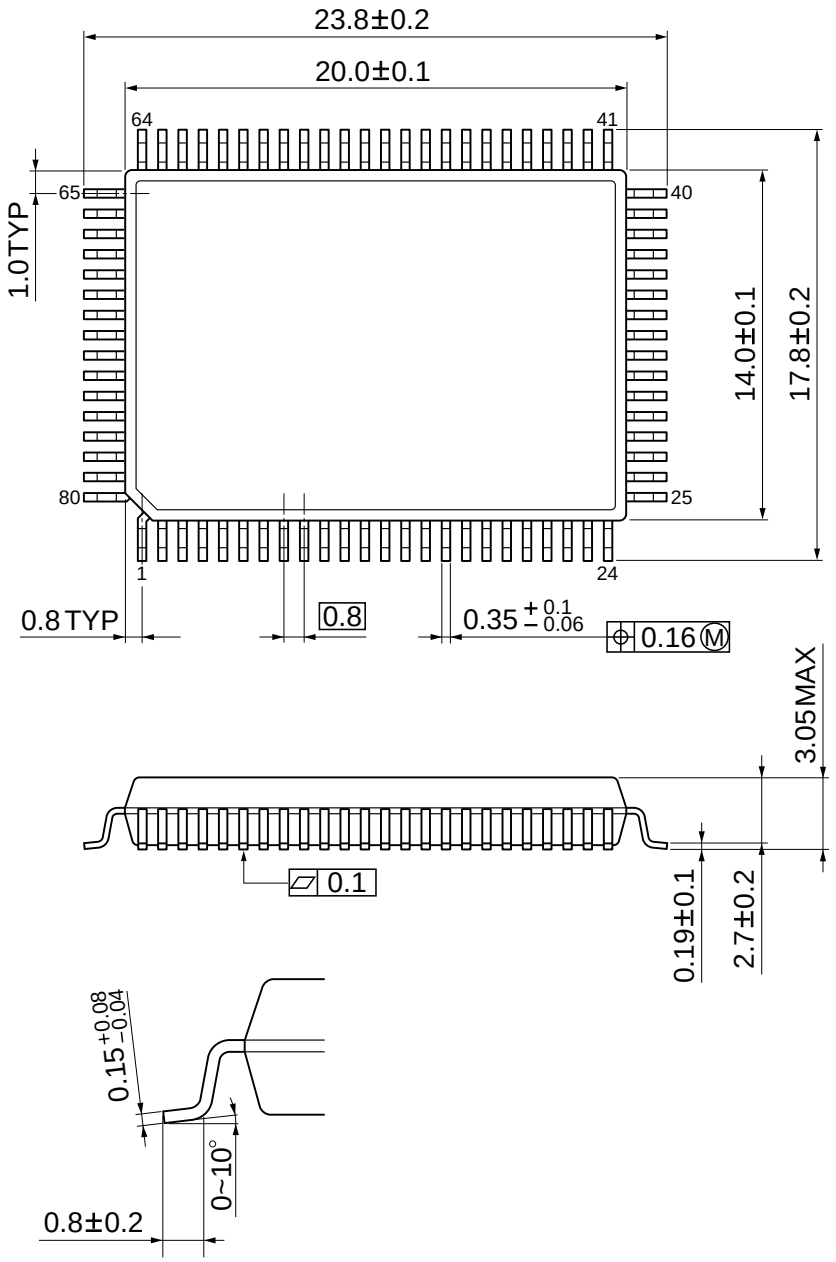
Unit: mm



Downloaded from Arrow.com.

QFP80-P-1420-0.80M Rev 02

Unit: mm



Downloaded from Arrow.com.

RESTRICTIONS ON PRODUCT USE

- Toshiba Corporation, and its subsidiaries and affiliates (collectively "TOSHIBA"), reserve the right to make changes to the information in this document, and related hardware, software and systems (collectively "Product") without notice.
- This document and any information herein may not be reproduced without prior written permission from TOSHIBA. Even with TOSHIBA's written permission, reproduction is permissible only if reproduction is without alteration/omission.
- Though TOSHIBA works continually to improve Product's quality and reliability, Product can malfunction or fail. Customers are responsible for complying with safety standards and for providing adequate designs and safeguards for their hardware, software and systems which minimize risk and avoid situations in which a malfunction or failure of Product could cause loss of human life, bodily injury or damage to property, including data loss or corruption. Before customers use the Product, create designs including the Product, or incorporate the Product into their own applications, customers must also refer to and comply with (a) the latest versions of all relevant TOSHIBA information, including without limitation, this document, the specifications, the data sheets and application notes for Product and the precautions and conditions set forth in the "TOSHIBA Semiconductor Reliability Handbook" and (b) the instructions for the application with which the Product will be used with or for. Customers are solely responsible for all aspects of their own product design or applications, including but not limited to (a) determining the appropriateness of the use of this Product in such design or applications; (b) evaluating and determining the applicability of any information contained in this document, or in charts, diagrams, programs, algorithms, sample application circuits, or any other referenced documents; and (c) validating all operating parameters for such designs and applications. **TOSHIBA ASSUMES NO LIABILITY FOR CUSTOMERS' PRODUCT DESIGN OR APPLICATIONS**.
- **PRODUCT IS NEITHER INTENDED NOR WARRANTED FOR USE IN EQUIPMENTS OR SYSTEMS THAT REQUIRE EXTRAORDINARILY HIGH LEVELS OF QUALITY AND/OR RELIABILITY, AND/OR A MALFUNCTION OR FAILURE OF WHICH MAY CAUSE LOSS OF HUMAN LIFE, BODILY INJURY, SERIOUS PROPERTY DAMAGE AND/OR SERIOUS PUBLIC IMPACT ("UNINTENDED USE").** Except for specific applications as expressly stated in this document, Unintended Use includes, without limitation, equipment used in nuclear facilities, equipment used in the aerospace industry, Class 3 medical devices, equipment used for automobiles, and military vehicles and munitions. **IF YOU USE PRODUCT FOR UNINTENDED USE, TOSHIBA ASSUMES NO LIABILITY FOR PRODUCT.** For details, please contact your TOSHIBA sales representative or contact us via our website.
- Do not disassemble, analyze, reverse-engineer, alter, modify, translate or copy Product, whether in whole or in part.
- Product shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable laws or regulations.
- The information contained herein is presented only as guidance for Product use. No responsibility is assumed by TOSHIBA for any infringement of patents or any other intellectual property rights of third parties that may result from the use of Product. No license to any intellectual property right is granted by this document, whether express or implied, by estoppel or otherwise.
- **ABSENT A WRITTEN SIGNED AGREEMENT, EXCEPT AS PROVIDED IN THE RELEVANT TERMS AND CONDITIONS OF SALE FOR PRODUCT, AND TO THE MAXIMUM EXTENT ALLOWABLE BY LAW, TOSHIBA (1) ASSUMES NO LIABILITY WHATSOEVER, INCLUDING WITHOUT LIMITATION, INDIRECT, CONSEQUENTIAL, SPECIAL, OR INCIDENTAL DAMAGES OR LOSS, INCLUDING WITHOUT LIMITATION, LOSS OF PROFITS, LOSS OF OPPORTUNITIES, BUSINESS INTERRUPTION AND LOSS OF DATA, AND (2) DISCLAIMS ANY AND ALL EXPRESS OR IMPLIED WARRANTIES AND CONDITIONS RELATED TO SALE, USE OF PRODUCT, OR INFORMATION, INCLUDING WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, ACCURACY OF INFORMATION, OR NONINFRINGEMENT.**
- Do not use or otherwise make available Product or related software or technology for any military purposes, including without limitation, for the design, development, use, stockpiling or manufacturing of nuclear, chemical, or biological weapons or missile technology products (mass destruction weapons). Product and related software and technology may be controlled under the applicable export laws and regulations including, without limitation, the Japanese Foreign Exchange and Foreign Trade Law and the U.S. Export Administration Regulations. Export and re-export of Product or related software or technology are strictly prohibited except in compliance with all applicable export laws and regulations.
- Please contact your TOSHIBA sales representative for details as to environmental matters such as the RoHS compatibility of Product. Please use Product in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. **TOSHIBA ASSUMES NO LIABILITY FOR DAMAGES OR LOSSES OCCURRING AS A RESULT OF NONCOMPLIANCE WITH APPLICABLE LAWS AND REGULATIONS.**

Downloaded from Arrow.com.

Downloaded from Arrow.com.